Contents

1. Aloha…….... 1
2. CSMA……. 6
3. CSMA/CD……..……..... 11
4. Token Bus……. 16
5. Token Ring……..... 21
6. Stop and Wait…….. 25
7. Stop and Wait with BER…… 28
8. Sliding Window Go Back N…….. 30
9. Sliding Window Go Back N with BER……... 33
10. Packet Transmission……. 35
11. File transfer …… 38
12. Appendix 1 Graph plot procedure……. 41
13. Appendix 2 Configuration menu details…….. 48
14. Appendix 3 Developing Lan-T application…….. 50

Experiment 1 – ALOHA

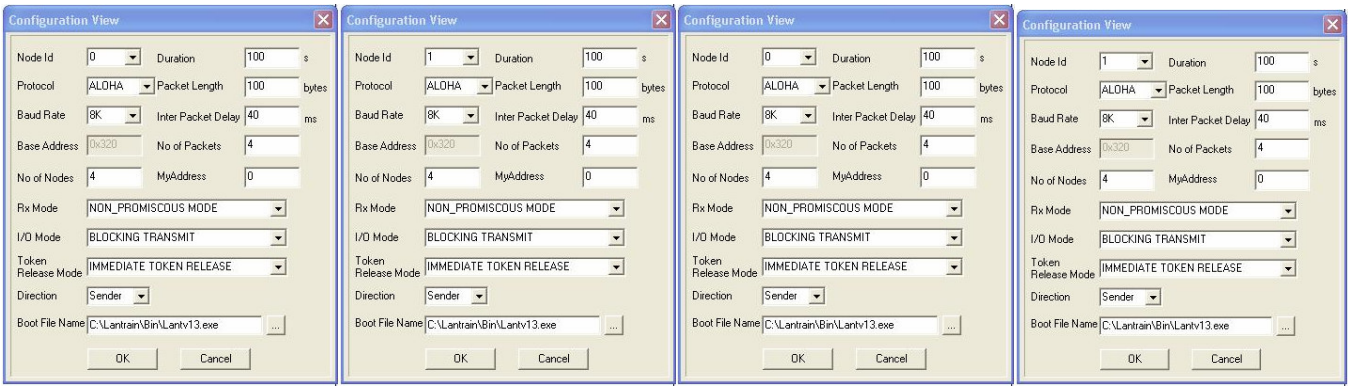
Aim:

Implement the *ALOHA* protocol for packet communication between a number of nodes connected to a common bus

Procedure:

1. Click on the **MAC** Experiment icon twice from the desktop on both PC's.
2. Click the Configuration button in the window in both the PC's.

PC 1 **PC 2**


Configuration View
Node Id 0 Duration 100 s
Protocol ALOHA Packet Length 100 bytes
Baud Rate 8K Inter Packet Delay 40 ms
Base Address 0x320 No of Packets 4
No of Nodes 4 MyAddress 0
Rx Mode NON_PROMISCOUS MODE
I/O Mode BLOCKING TRANSMIT
Token Release Mode IMMEDIATE TOKEN RELEASE
Direction Sender
Boot File Name C:\Lantrain\Bin\Lantv13.exe
OK Cancel
Configuration View
Node Id 1 Duration 100 s
Protocol ALOHA Packet Length 100 bytes
Baud Rate 8K Inter Packet Delay 40 ms
Base Address 0x320 No of Packets 4
No of Nodes 4 MyAddress 0
Rx Mode NON_PROMISCOUS MODE
I/O Mode BLOCKING TRANSMIT
Token Release Mode IMMEDIATE TOKEN RELEASE
Direction Sender
Boot File Name C:\Lantrain\Bin\Lantv13.exe
OK Cancel
Configuration View
Node Id 0 Duration 100 s
Protocol ALOHA Packet Length 100 bytes
Baud Rate 8K Inter Packet Delay 40 ms
Base Address 0x320 No of Packets 4
No of Nodes 4 MyAddress 0
Rx Mode NON_PROMISCOUS MODE
I/O Mode BLOCKING TRANSMIT
Token Release Mode IMMEDIATE TOKEN RELEASE
Direction Sender
Boot File Name C:\Lantrain\Bin\Lantv13.exe
OK Cancel
Configuration View
Node Id 1 Duration 100 s
Protocol ALOHA Packet Length 100 bytes
Baud Rate 8K Inter Packet Delay 40 ms
Base Address 0x320 No of Packets 4
No of Nodes 4 MyAddress 0
Rx Mode NON_PROMISCOUS MODE
I/O Mode BLOCKING TRANSMIT
Token Release Mode IMMEDIATE TOKEN RELEASE
Direction Sender
Boot File Name C:\Lantrain\Bin\Lantv13.exe
OK Cancel

Setting the Configuration menu:

PC 1		PC 2	
Node id	**0** on config menu 1 and **1** on config menu 2	**Node id**	**0** on config menu 1 and **1** on config menu 2
Protocol	ALOHA	**Protocol**	ALOHA
Baud Rate	8Kbps (At both the config menu and NEU)	**Baud Rate**	8Kbps (At both the config menu and NEU)
Duration	100s	**Duration**	100s
Packet Length	100 bytes	**Packet Length**	100 bytes
Bit Delay	0(at NEU)	**Bit Delay**	0(at NEU)
Direction	Sender	**Direction**	Sender

Note: All the nodes have to be configured as 'Senders'. Set the topology as 'Bus'.

$$G = \frac{N * P}{C * t_a}$$

G is the generated load in the network.

N is the number of nodes participating in the network. For example, let us say that 4 nodes (using 2 computers)

P is the packet length expressed in bits; say 100 bytes (800 bits).

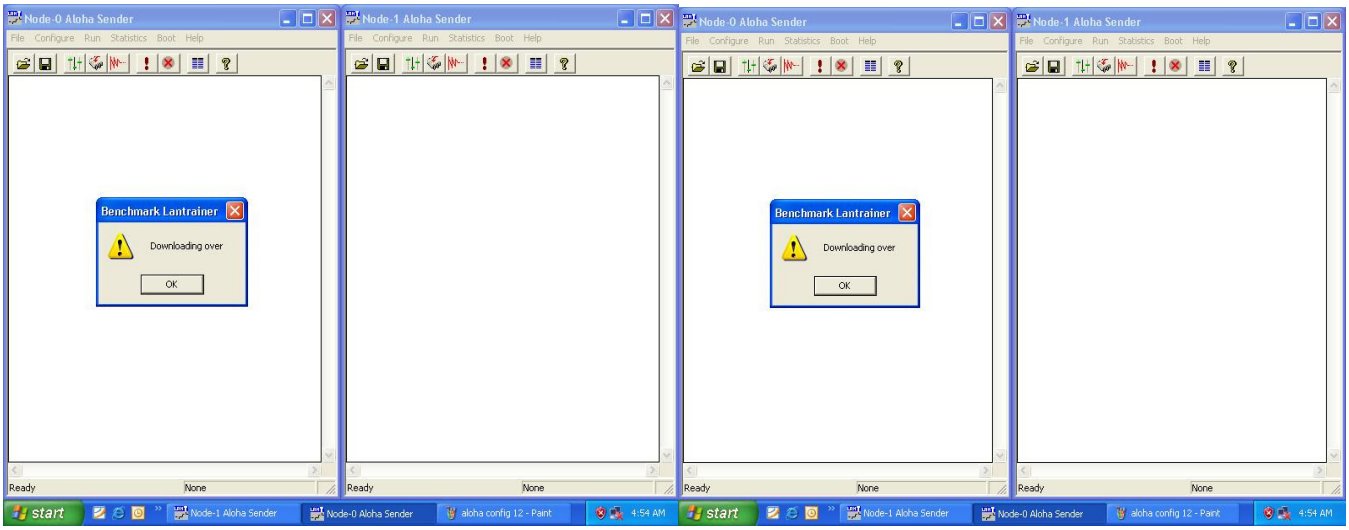
C is the data rate normally set as 8kbs, which is selected in the NEU.

$\mathbf{t_a}$ is the inter packet delay expressed in seconds; the time interval between two consecutive packets generated.

So, lets assume $\mathbf{t_a}$= 40 milliseconds and substitute the above mentioned parameters in the Equation A which leads to G = 10. Like wise assume various values for $\mathbf{t_a}$ to generate offer loads in the range of 0.1 to 10. Substitute the value of $\mathbf{t_a}$ in the configuration menu.

3. Click **OK** button and Download the driver to the NIU using the BOOT button command. Booting from any one of the applications is enough.

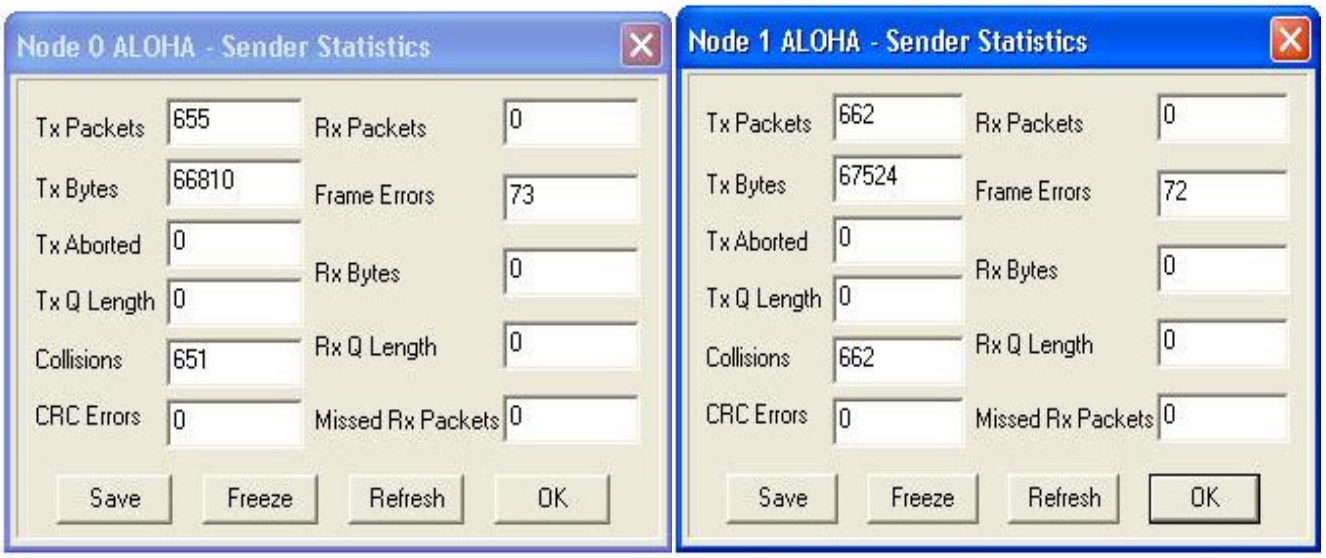
PC 1 **PC 2**



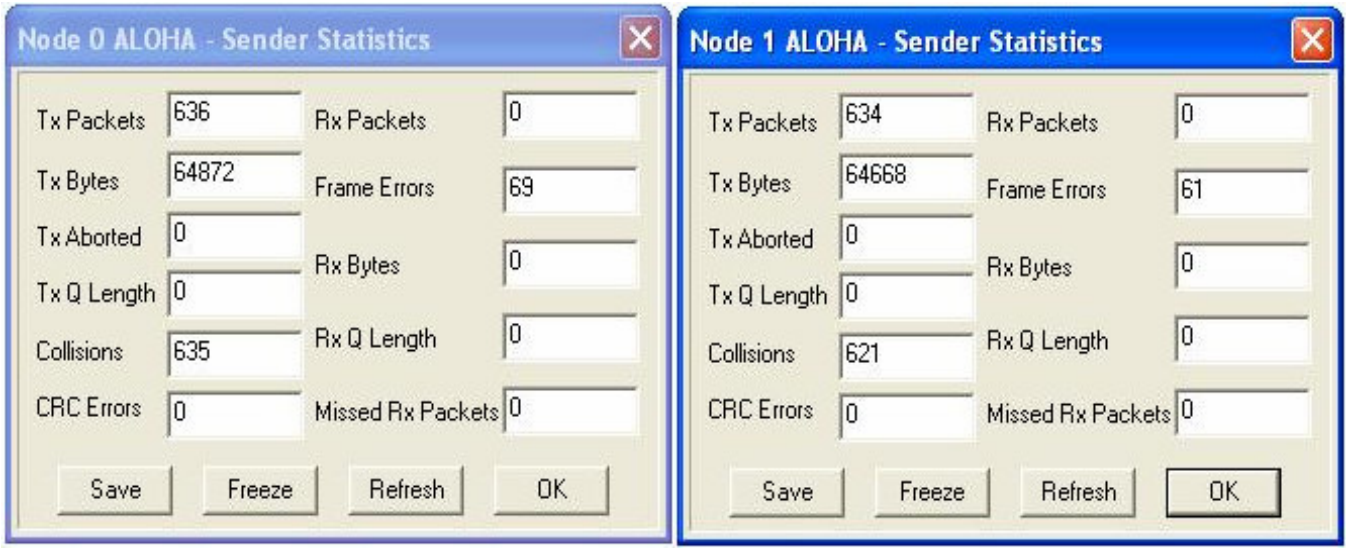
4. Run the experiment by clicking button or by choosing **RUN → Start** from each application.

5. View the statistics window for results. To view the statistics window click on button. Only Tx packets and collision count are taken into account for MAC calculation.

PC 1



PC 2



6. Note down the readings once the experiment is completed.

7. Repeat the above steps for various values of $\mathbf{t_a}$.

8. Calculate the Practical offered load from the below given formula and plot the graph between the practical Offered load and Throughput.

Note: You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template.

9. Repeat the experiment for various values of Packet length, Node, Data rate.

Calculation of Practical Throughput (X) from the obtained readings:

Successfully transmitted packet by a node = Tx Packets - Collision Count

$$X = \frac{(\text{Sum of Successfully Tx packet in all the nodes} * \text{Packet Length} * 8)}{(\text{Duration of Experiment} * \text{Data rate})}$$

Calculation of Theoretical Throughput:

$$X = Ge^{-2G}$$

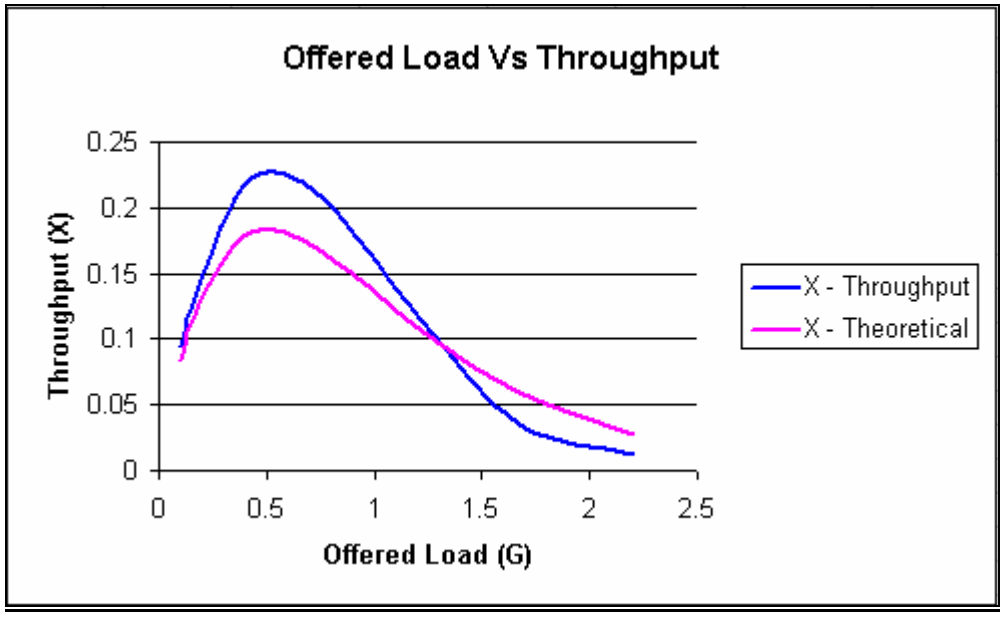
Calculation of Practical Offered load:

$$G = \frac{(\text{Sum of Transmitted packets in all the nodes} * \text{Packet Length} * 8)}{(\text{Duration of Experiment} * \text{Data rate})}$$

Model tabulation:

IPD	Tx1	Tx 2	Tx 3	Tx 4	G – Practical Offered Load	X – Practical Throughput	Theoretical Throughput
4000	23	19	25	16	0.101	0.083	0.08
2000	36	35	38	33	0.2	0.142	0.13
800	54	48	43	60	0.444	0.205	0.18
400	52	41	39	45	0.766	0.177	0.14
200	18	26	11	19	1.293	0.074	0.04
100	10	1	4	10	1.82	0.025	0.00
40	4	0	0	6	2.484	0.01	0.00

Model Graph:



Experiment 2 – CSMA

Aim:

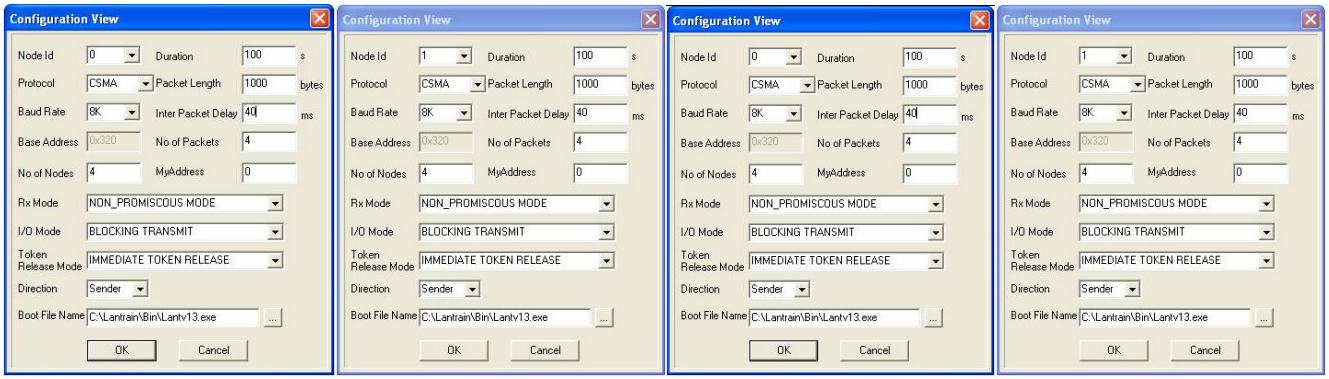
Implement the *CSMA* protocol for packet communication between a number of nodes connected to a common bus

Procedure:

1. Click on the **MAC** Experiment icon

twice from the desktop on both PC's.
2. Click the Configuration button in the window in both the PC's.

PC 1 **PC 2**



Configuration menu:

PC 1		PC 2	
Node id	**0** on config menu 1 and **1** on config menu 2	**Node id**	**0** on config menu 1 and **1** on config menu 2
Protocol	CSMA	**Protocol**	CSMA
Baud Rate	8Kbps (At both the config menu and NEU)	**Baud Rate**	8Kbps (At both the config menu and NEU)
Duration	100s	**Duration**	100s
Packet Length	100 bytes	**Packet Length**	100 bytes
Bit Delay	10 (at NEU)	**Bit Delay**	10 (at NEU)
Direction	Sender	**Direction**	Sender

Note: All the nodes have to be configured as 'Senders'. Set the topology as 'Bus'.

$$G = \frac{N * P}{C * t_a}$$

G is the generated load in the network.

N is the number of nodes participating in the network. For example, let us say that 4 nodes (using 2 computers)

P is the packet length expressed in bits; say 100 bytes (800 bits).

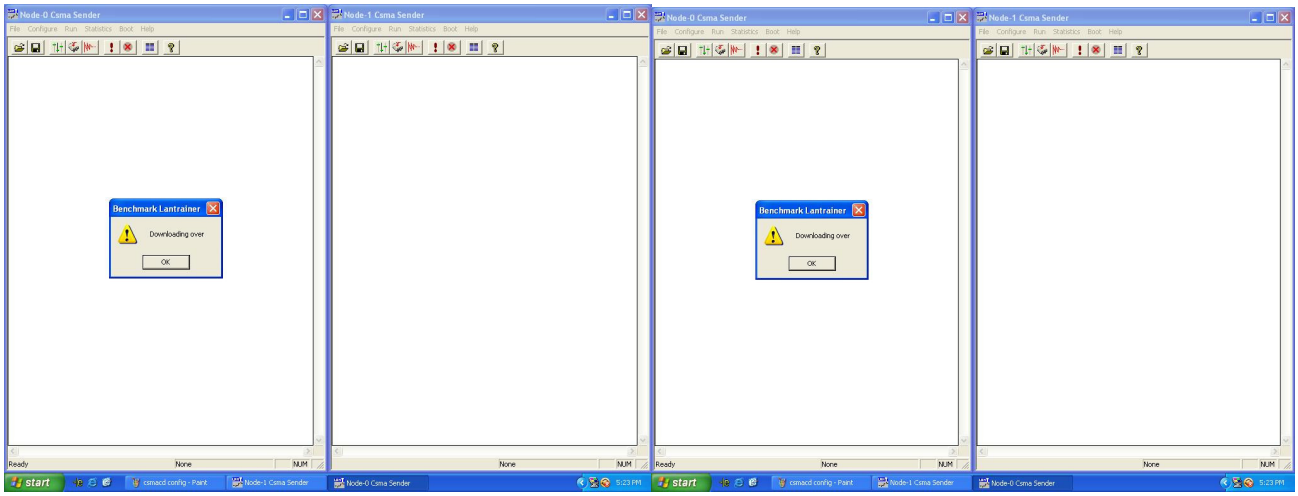
C is the data rate normally set as 8kbs, which is selected in the NEU.

$\mathbf{t_a}$ is the inter packet delay expressed in seconds; the time interval between two consecutive packets generated.

So, lets assume t_a= 40 milliseconds and substitute the above mentioned parameters in the Equation A which leads to G = 10. Like wise assume various values of $\mathbf{t_a}$ to generate offer loads in the range of 0.1 to 10. Substitute the $\mathbf{t_a}$ in the configuration menu.

3. Click **OK** button and Download the driver to the NIU using the BOOT button command. Booting from any one of the applications is enough.

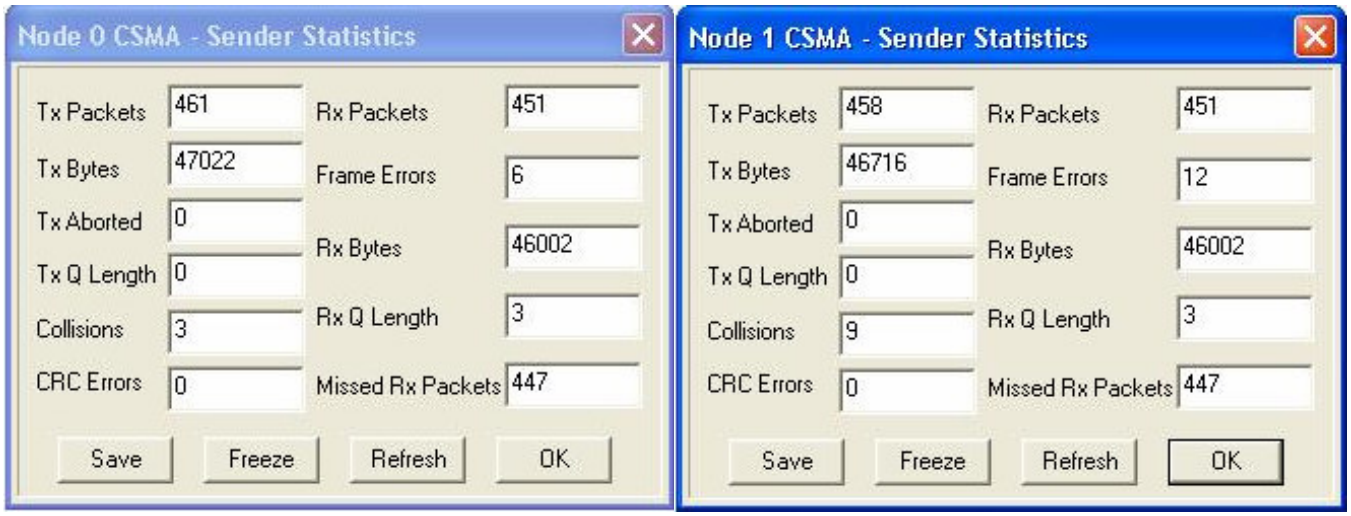
PC 1 **PC 2**



4. Run the experiment by clicking button or by choosing RUN → Start from each application.

5. View the statistics window for results. To view the statistics window click on button. Only Tx packets and collision count are taken into account for MAC calculation.

PC 1



PC 2

6. Note down the readings once the experiment is completed.

7. Repeat the above steps for various values of $\mathbf{t_a}$.

8. Calculate the Practical offered load from the below given formula and plot the graph between the practical Offered load and Throughput.

Note: You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template.

9. Repeat the experiments for various values of Packet length, Node, Data rate and Bit delay.

Calculation of Practical Throughput (X) from the obtained readings:

Successfully transmitted packet by a node = Tx Packets - Collision Count

$$X = \frac{\text{(Sum of Successfully Tx packet in all the nodes * Packet Length * 8)}}{\text{(Duration of Experiment * Data rate)}}$$

Calculation of Theoretical Throughput:

$$X = \frac{G(1+G+aG(1+G+aG/2))e^{-G(1+2a)}}{G(1+2a)-(1-e^{-aG})+(1+aG)e^{-G(1+a)}}$$

a = (end to end bit delay in bits) / (Packet length in bits) = (bit delay*N) / (P)

Calculation of Practical Offered load:

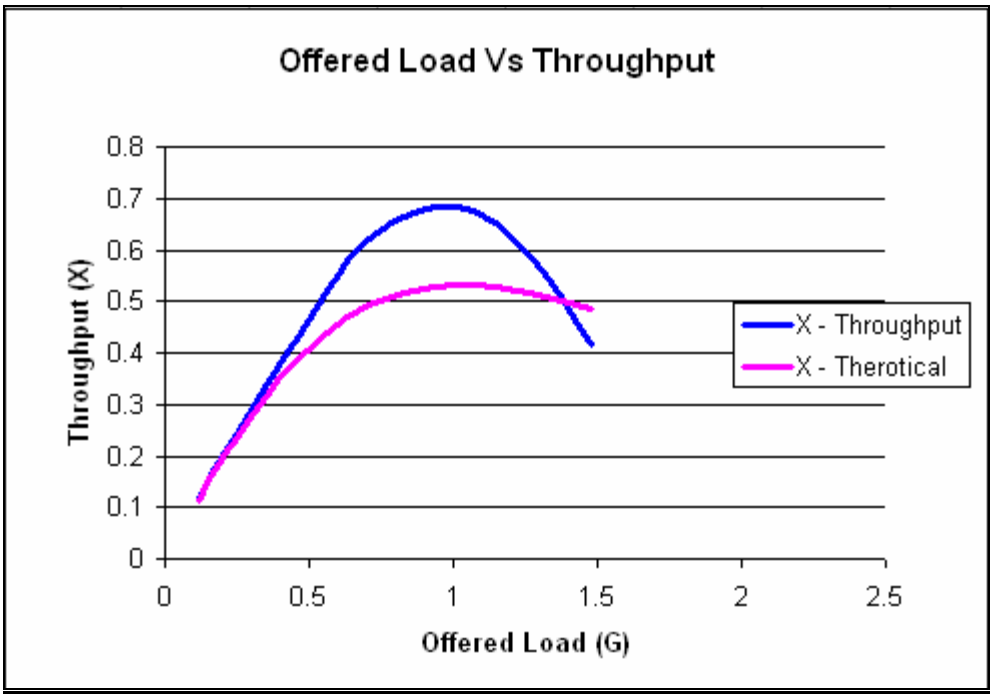
$$G = \frac{\text{(Sum of Transmitted packets in all the nodes * Packet Length * 8)}}{\text{(Duration of Experiment * Data rate)}}$$

Model tabulation:

For bit delay = 1

IPD	Tx1	Tx 2	Tx 3	Tx 4	G – Practical Offered Load	X – Practical Throughput	Theoretical Throughput
4000	29	28	29	28	0.116	0.114	0.11
2000	49	50	49	49	0.197	0.197	0.19
800	107	93	99	101	0.436	0.400	0.37
400	151	124	134	144	0.725	0.553	0.49
200	166	156	155	165	1.076	0.642	0.53
100	137	103	126	116	1.388	0.482	0.52
40	12	12	7	37	1.824	0.068	0.48

Model Graph:



Experiment 3 – CSMA/CD

Aim:

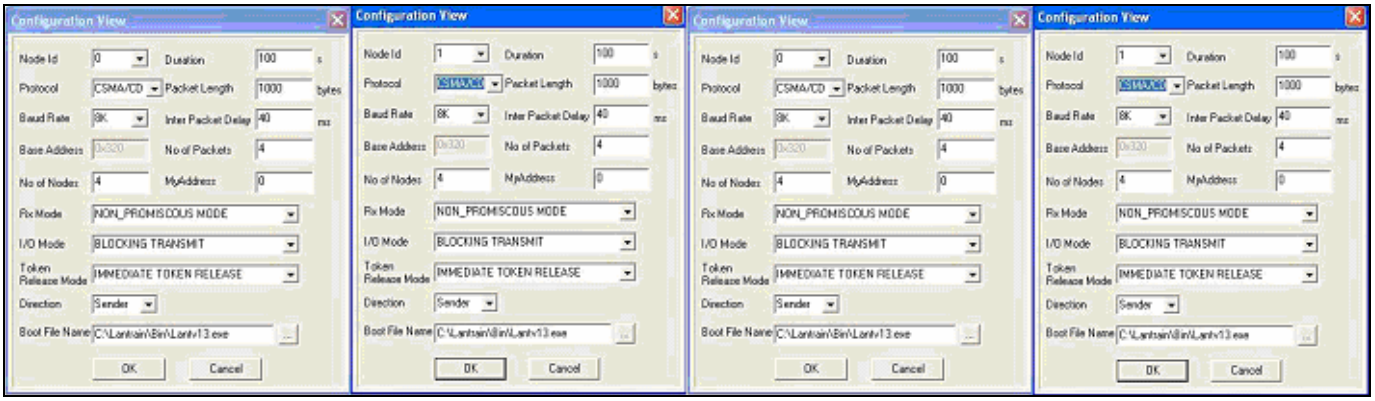
Implement the *CSMA/CD* protocol for packet communication between a number of nodes connected to a common bus

Procedure:

1. Click on the **MAC** Experiment icon

twice from the desktop on both PC's.
2. Click the Configuration button in the window in both the PC's.

PC 1 **PC 2**



Configuration menu:

PC 1		PC 2	
Node id	**0** on config menu 1 and **1** on config menu 2	**Node id**	**0** on config menu 1 and **1** on config menu 2
Protocol	CSMA/CD	**Protocol**	CSMA/CD
Baud Rate	8Kbps (At both the config menu and NEU)	**Baud Rate**	8Kbps (At both the config menu and NEU)
Duration	100s	**Duration**	100s
Packet	1000 bytes	**Packet**	1000 bytes
Bit Delay	0(at NEU)	**Bit Delay**	0(at NEU)
Direction	Sender	**Direction**	Sender

Note: All the nodes have to be configured as 'Senders'. Set the topology as 'Bus'.

$$G = \frac{N * P}{C * t_a}$$

G is the generated load in the network.

N is the number of nodes participating in the network. For example, let us say that 4 nodes (using 2 computers)

P is the packet length expressed in bits; say 100 bytes (800 bits).

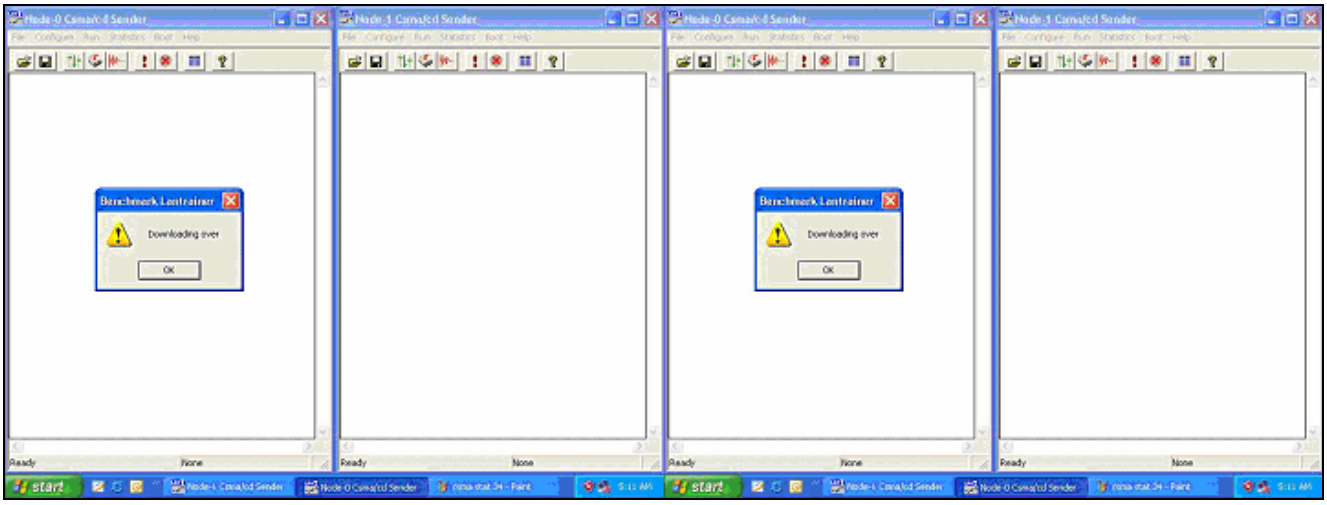
C is the data rate normally set as 8kbs, which is selected in the NEU.

$\mathbf{t_a}$ is the inter packet delay expressed in seconds; the time interval between two consecutive packets generated.

So, lets assume $\mathbf{t_a}$= 40 milliseconds and substitute the above mentioned parameters in the Equation A which leads to G = 10. Like wise assume various values of $\mathbf{t_a}$ to generate offer loads in the range of 0.1 to 10. Substitute the value of $\mathbf{t_a}$ in the configuration menu.

3. Click **OK** button and Download the driver to the NIU using the BOOT button command. Booting from any one of the applications is enough.

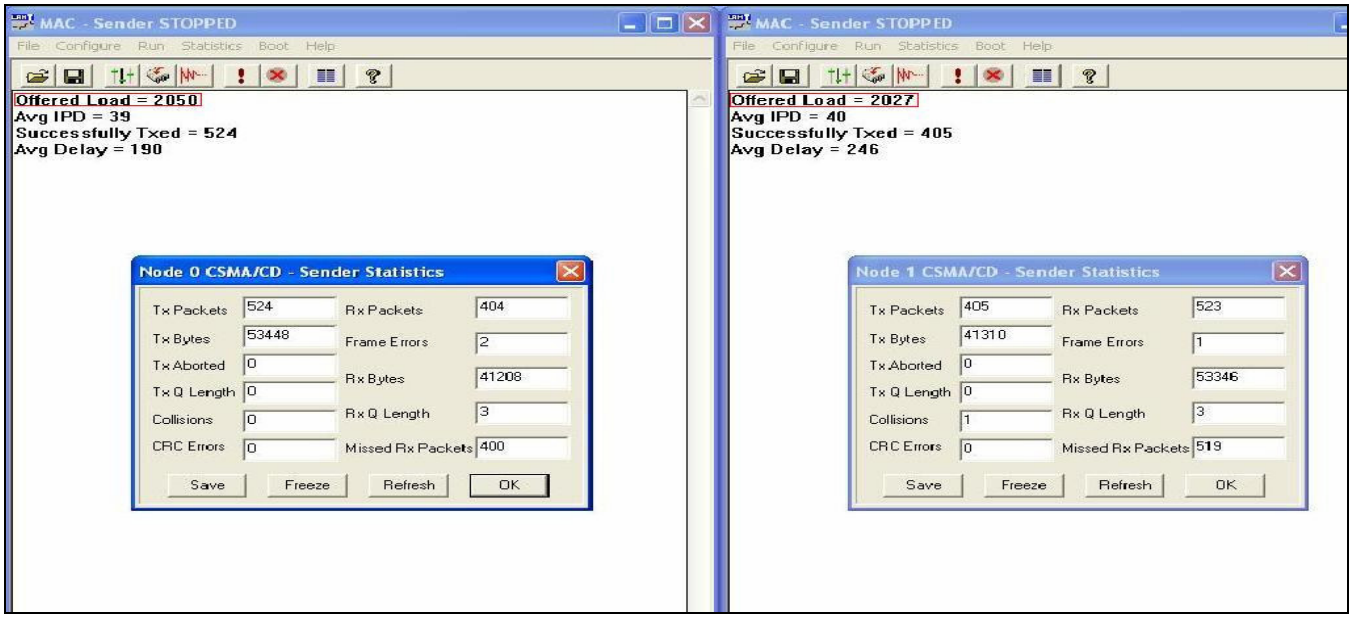
PC 1 **PC 2**



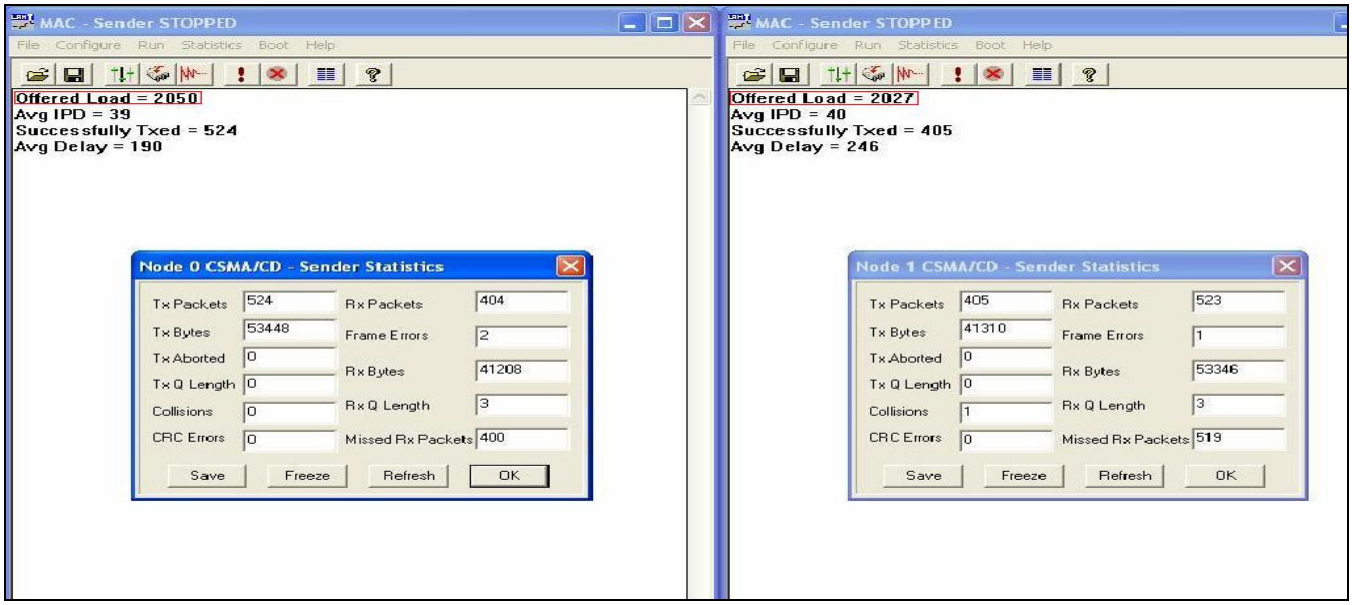
4. Run the experiment by clicking button or by choosing RUN → Start from each application.

5. View the statistics window for results. To view the statistics window click on button. Only Tx packets and successfully transmitted packets are taken into account for CSMA/CD calculation.

PC 1



PC 2



6. Note down the readings once the experiment is completed.

7. Repeat the above steps for various values of $\mathbf{t_a}$.

8. Calculate the Practical offered load from the below given formula and plot the graph between the practical Offered load and Throughput.

Note: You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template.

9. Repeat the experiment for various values of Packet length, Node, Data rate.

Calculation of Practical Throughput (X) from the obtained readings:

$$X = \frac{\text{(Sum of Successfully Tx packet in all the nodes * Packet Length * 8)}}{\text{(Duration of Experiment * Data rate)}}$$

Calculation of Practical Offered load:

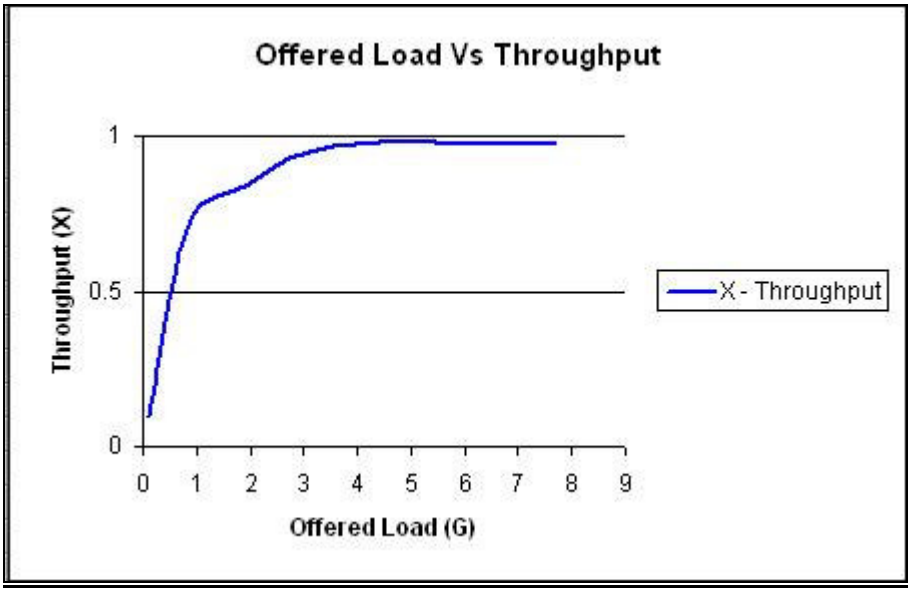
$$G = \frac{\text{(Sum of Offered load count in all the nodes * Packet Length * 8)}}{\text{(Duration of Experiment * Data rate)}}$$

*** offered load count and successfully transmitted packet count is displayed in the main window of CSMA/CD experiment.**

Model tabulation:

IPD	Tx1	Tx 2	Tx 3	Tx 4	G – Practical Offered Load	X – Practical Throughput
4000	27	25	24	24	0.101	0.1
2000	49	49	49	49	0.208	0.196
800	118	116	119	116	0.5	0.469
400	187	189	191	191	0.976	0.758
200	117	133	290	300	1.904	0.84
100	472	463	13	18	3.555	0.966
40	466	458	20	28	7.687	0.972

Model Graph:



Experiment 4 - TOKEN BUS

Aim:

To implement the token passing access in BUS-LAN

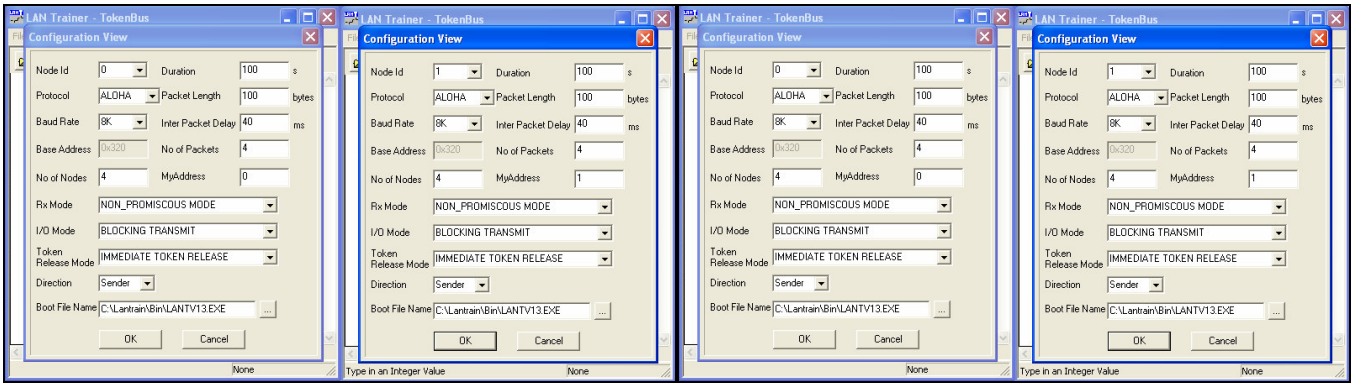
Procedure:

1. Click on the Token Bus icon

twice from the desktop.

2. Click the Configuration button in the window in both the PC's.

PC 1 – Sender **PC – 2 Sender**



Setting the configuration menu:

PC 1		PC 2	
Node id	**0** on config menu 1 and **1** on config menu 2	**Node id**	**0** on config menu 1 and **1** on config menu 2
Protocol	ALOHA	**Protocol**	ALOHA
Baud Rate	8Kbps (At both the config menu and NEU)	**Baud Rate**	8Kbps (At both the config menu and NEU)
Duration	100s	**Duration**	100s
Packet Length	100 bytes	**Packet Length**	100 bytes
My Address	**0** on config menu 1 and **1** on menu 2	**My Address**	**2** on config menu 1 and **3** on menu 2
Bit Delay	0(at NEU)	**Bit Delay**	0(at NEU)
Direction	Sender	**Direction**	Sender

Notes

1: If you connect two PC's and configured four nodes then set the **My Address** as 0 to 3 in all four nodes, if you connect three PCs and configured six nodes then set the **My Address** as 0 to 5 in all six nodes.

2: Start running the experiment from the lowest priority node (i.e., from My Address 3 in case of four nodes and 5 in the case of six nodes)

3: No of Nodes has to be set as 4 when two PCs are connected and 6 when three PCs are connected.

$$G = \frac{N * P}{C * t_a}$$

G is the generated load in the network.

N is the number of nodes participating in the network. For example, let us say that 4 nodes (using 2 computers)

P is the packet length expressed in bits; say 100 bytes (800 bits).

C is the data rate normally set as 8kbs, which is selected in the NEU.

$\mathbf{t_a}$ is the inter packet delay expressed in seconds; the time interval between two consecutive packets generated.

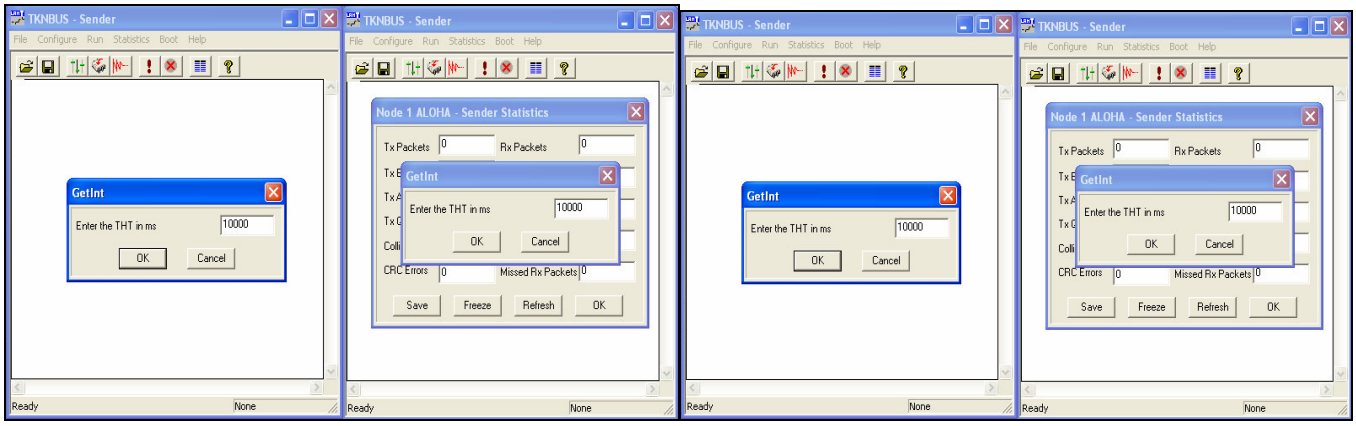
So, lets assume $\mathbf{t_a}$= 40 milliseconds and substitute the above mentioned parameters in the Equation A which leads to G = 10. Like wise assume various values of $\mathbf{t_a}$ to generate offer loads in the range of 0.1 to 10. Substitute the value of $\mathbf{t_a}$ in the configuration menu.

3. Click **OK** button and Download the driver to the NIU using the BOOT button command. Booting from any one of the applications is enough.

4. Run the experiment by clicking button or by choosing RUN → Start from each application. Run the all the experiments at the same time.

5. Set the Token Holding Time (**THT**) (say 10000 ms).

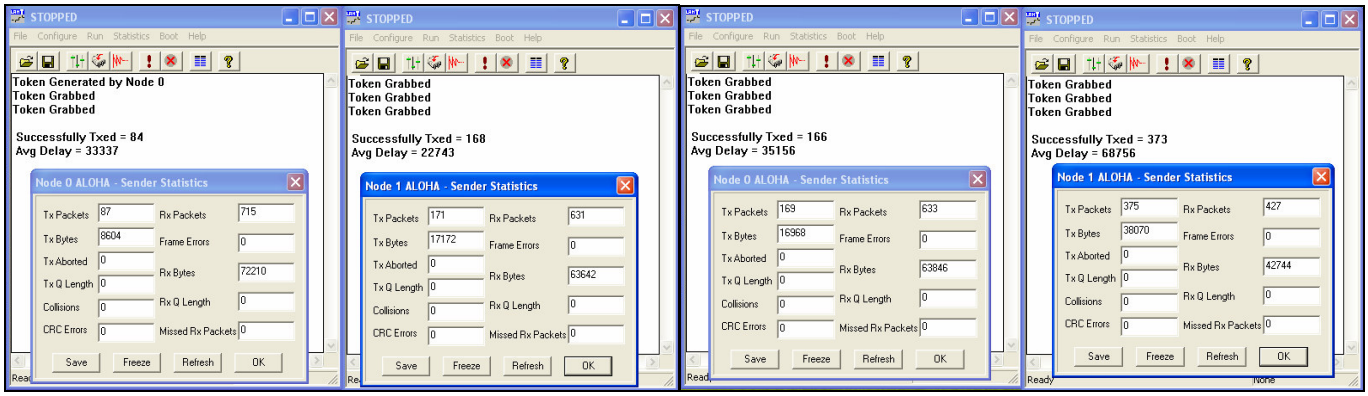
PC 1 **PC 2**



Note: While you do this **THT** window pops up, enter the THT time in all nodes and press the **OK** button first in the node, which has highest value of **My Address**.

6. View the statistics window for results. To view the statistics window click on button.

PC 1 **PC2**



7. Note down the readings once the experiment is completed. sw

8. Repeat the above steps for various values of $\mathbf{t_a}$.

9. Calculate the Practical offered load from the below given formula and plot the graph between the practical Offered load and Throughput.

Note: You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template.

10. Repeat the experiment for various values of Packet length, Node, Data rate.

11. Repeat the above steps, while running the experiment set the **BER to 10^{-2}** in the **NEU** or try to stop one of the nodes and observe the behavior and explain the same.

Calculation of Practical Throughput (X) from the obtained readings:

$$X = \frac{\text{(Sum of Tx packet in all the nodes * Packet Length * 8)}}{\text{(Duration of Experiment * Data rate)}}$$

Calculation of the Offered load:

$$G = \frac{N * P}{C * t_a}$$

G – Offered load

N – Number of nodes

P – Packet length in bits

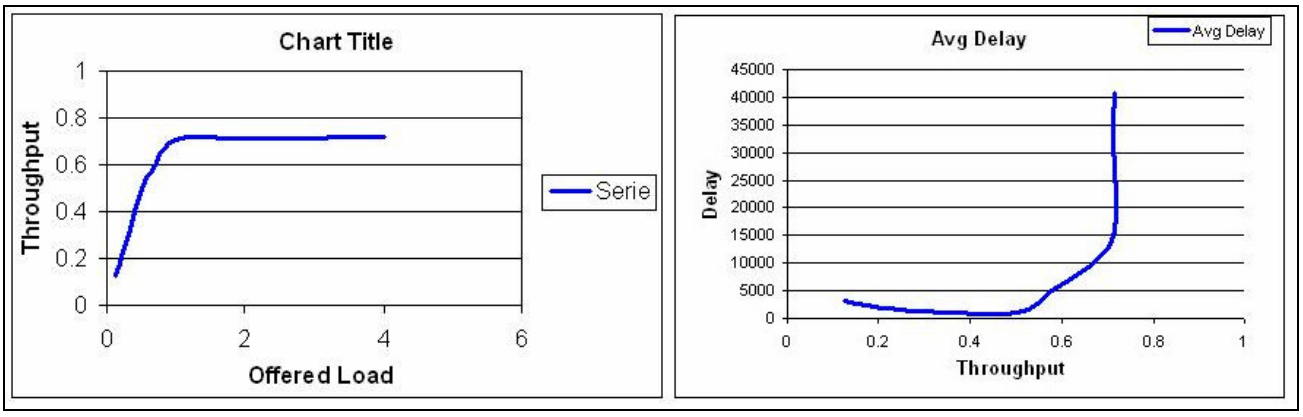
C – Data rate in bits/sec

t_a – Inter packet delay in millisecs.

Model Tabulation:

IPD	Tx 1	Tx 2	Tx 3	Tx 4	G - Offered Load	X –Practical Throughput	Avg Delay
16000	7	7	9	5	0.25	0.28	12205.5
8000	11	11	11	13	0.5	0.46	7787.75
4000	18	21	26	19	1	0.84	11265
2000	24	20	21	24	2	0.89	28604.75
1000	20	18	30	20	4	0.88	38216

Model Graph



Experiment 5 - TOKEN RING

Aim:

To implement the token passing access in RING-LAN

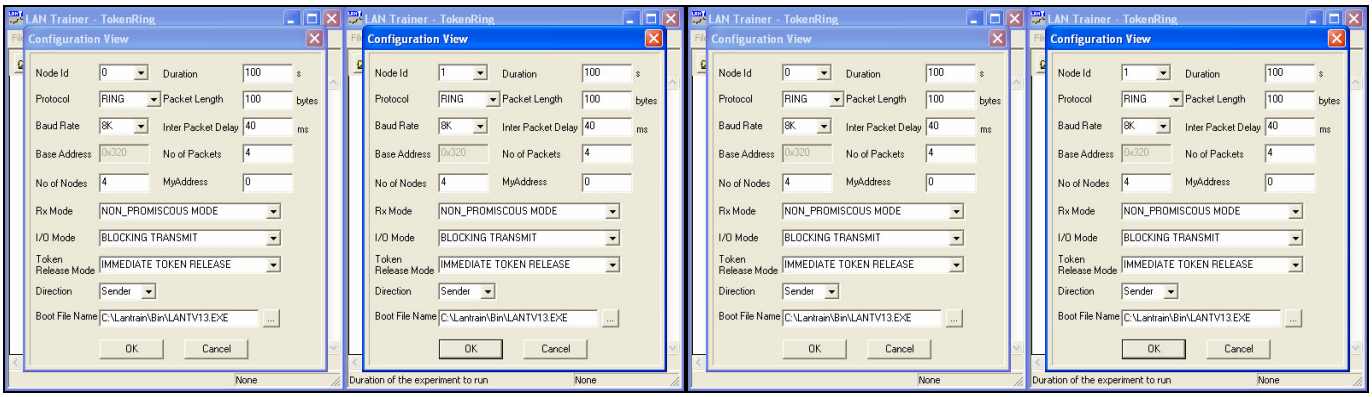
Procedure:

1. Click on the Token Bus icon

twice from the desktop.

2. Click the Configuration button in the window in both the PC's.

PC-1 **PC–2**



Setting the configuration menu:

PC 1		PC 2	
Node id	**0** on config menu 1 and **1** on config menu 2	**Node id**	**0** on config menu 1 and **1** on config menu 2
Protocol	RING	**Protocol**	RING
Baud Rate	8Kbps (At both the config menu and NEU)	**Baud Rate**	8Kbps (At both the config menu and NEU)
Duration	100s	**Duration**	100s
Packet Length	1000 bytes	**Packet Length**	1000 bytes
Bit Delay	0(at NEU)	**Bit Delay**	0(at NEU)
Direction	Sender	**Direction**	Sender

.

$$G = \frac{N * P}{C * t_a}$$

G is the generated load in the network.

N is the number of nodes participating in the network. For example, let us say that 4 nodes (using 2 computers)

P is the packet length expressed in bits; say 100 bytes (800 bits).

C is the data rate normally set as 8kbs, which is selected in the NEU.

$\mathbf{t_a}$ is the inter packet delay expressed in seconds; the time interval between two consecutive packets generated.

So, lets assume $\mathbf{t_a}$= 40 milliseconds and substitute the above mentioned parameters in the Equation A which leads to G = 10. Like wise assume various values of $\mathbf{t_a}$ to generate offer loads in the range of 0.1 to 10. Substitute the value of $\mathbf{t_a}$ in the configuration menu.

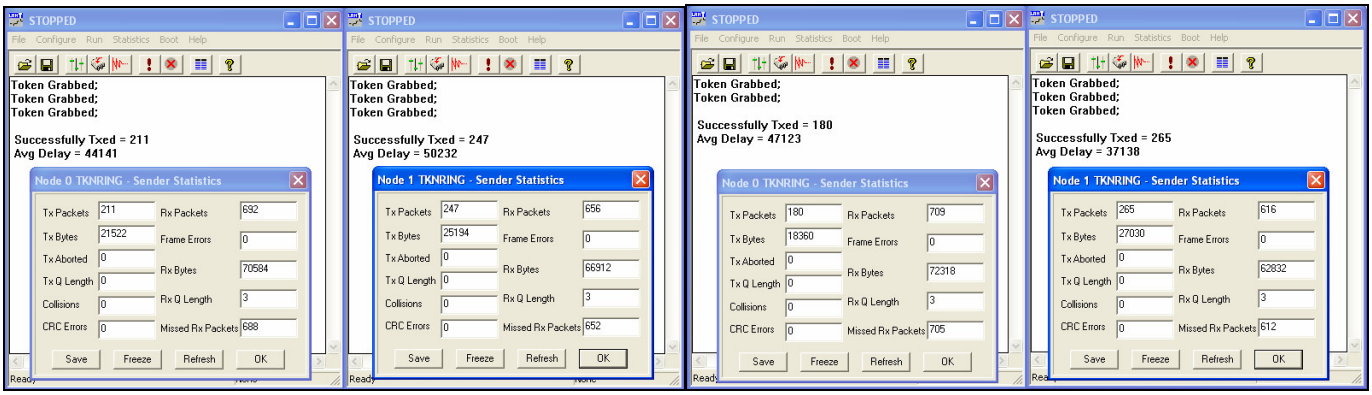
3. Click **OK** button and Download the driver to the NIU using the BOOT button command. Booting from any one of the applications is enough.

4. Run the experiment by clicking button or by choosing RUN → Start from each application. Run the all the experiments at the same time.

5. Set the Token Holding Time (**THT**) (say 10000 ms).

6. View the statistics window for results. To view the statistics window click on button.

PC 1 **PC2**



7. Note down the readings once the experiment is completed.

8. Repeat the above steps for various values of $\mathbf{t_a}$.

9. Calculate the Practical offered load from the below given formula and plot the graph between the practical Offered load and Throughput.

Note: You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template.

10. Repeat the experiments for various values of Packet length, Node, Data rate.

11. Repeat the above steps, while running the experiment set the **BER to 10^{-2}** in the **NEU** or try to stop one of the nodes and observe the behavior and explain the same.

Calculation of Practical Throughput (X) from the obtained readings:

$$X = \frac{\text{(Sum of Tx packet in all the nodes * Packet Length * 8)}}{\text{(Duration of Experiment * Data rate)}}$$

Calculation of the Offered load:

$$G = \frac{N * P}{C * t_a}$$

G – Offered load

N – Number of nodes

P – Packet length in bits

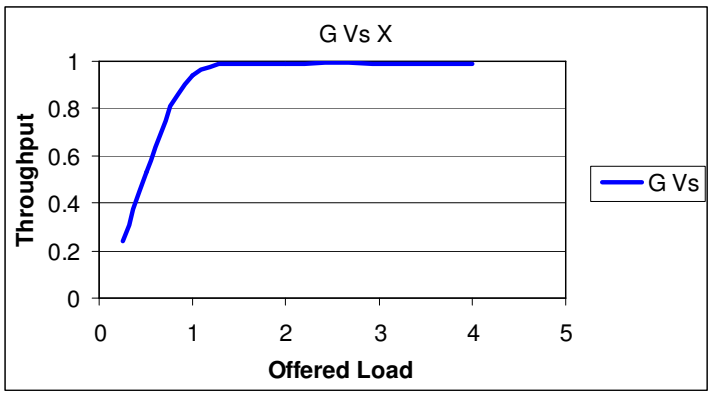
C – Data rate in bits/sec

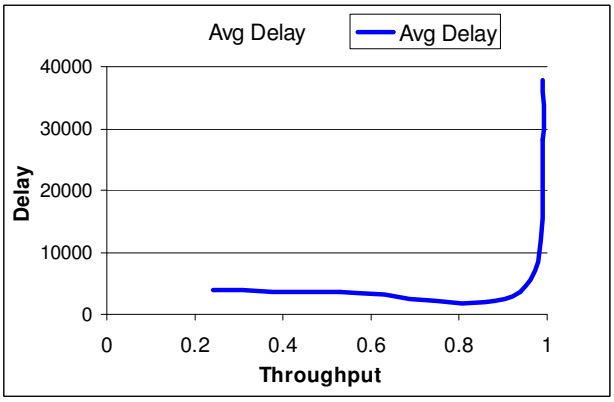
t_a – Inter packet delay in millisecs.

Model Tabulation:

IPD	Tx Node1	Tx Node2	Tx Node3	Tx Node4	G - Offered Load	X - Throughput	Avg Delay
16000	6	6	6	6	0.25	0.24	3779.75
8000	12	13	14	14	0.5	0.53	3708.25
4000	19	19	28	28	1	0.94	3658.5
2000	22	22	30	25	2	0.99	28270.75
1000	30	27	21	21	4	0.99	37720.25

Model Graph





Experiment 6 – STOP & WAIT

Aim:

Provide reliable data transfer between two nodes over an unreliable network using the stop-and-wait protocol

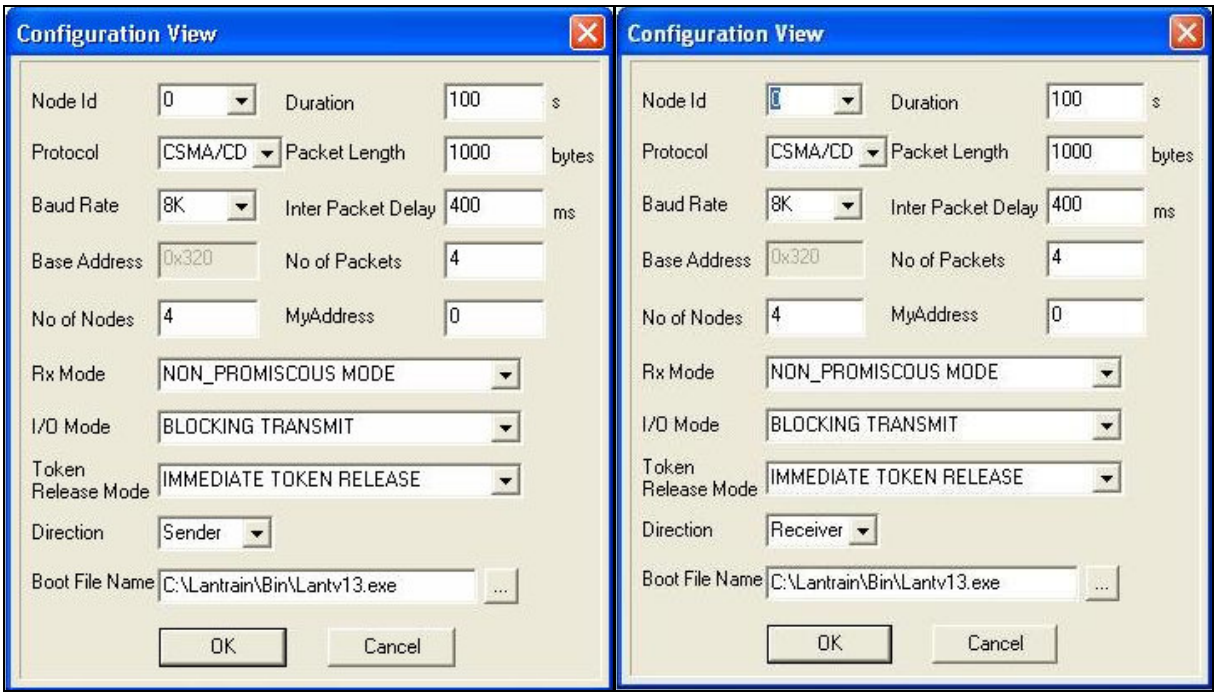
Procedure:

1. Click on the **Stop & Wait** icon

from the desktop on both PCs.

2. Click the Configuration button in the window in both the Pc's.

PC 1 SENDER **PC 2 RECEIVER**



Setting the configuration menu:

PC 1		PC 2	
Node id	**0**	**Node id**	**0**
Protocol	CSMA/CD	**Protocol**	CSMA/CD
Baud Rate	8Kbps (At both the config menu and NEU)	**Baud Rate**	8Kbps (At both the config menu and NEU)
Duration	100s	**Duration**	100s
Packet Length	1000 bytes	**Packet Length**	1000 bytes
Bit Delay	0(at NEU)	**Bit Delay**	0(at NEU)
Direction	Sender	**Direction**	Receiver

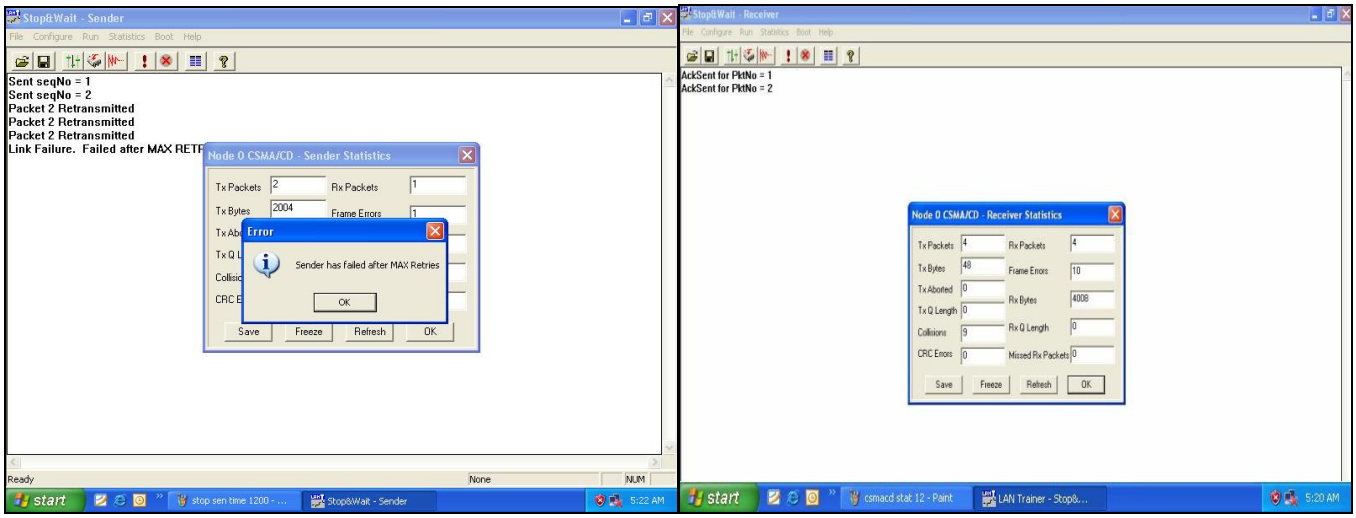
3. Set the **Inter Packet Delay** to 400msecs

4. Click **OK** button and Download the driver to the NIU using the BOOT button command. Booting from any one of the applications is enough.

5. Run the experiment by clicking button or by choosing RUN → Start from each application.

6. Set the Timeout Value to **1500** ms

PC 1 SENDER **PC 2 RECEIVER**



7. Note down the no of successfully Transmitted Packets.

8. Repeat the above steps for various **time out** values and plot the graph between timeout Value & Throughput. Find the optimum timeout value from the plot.

Note: You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template

9. Explain why the throughput is less compared to CSMACD protocol.

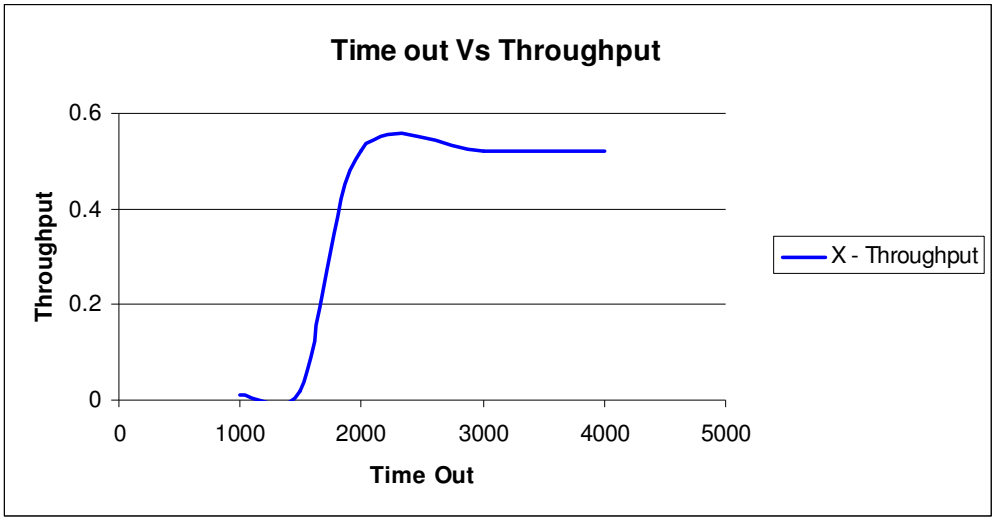
Calculation of Practical Throughput:

$$X = \frac{\text{(Sum of Successfully Tx packets * Packet Length * 8)}}{\text{(Duration of Experiment * Data rate)}}$$

Model Tabulation:

Time out value in ms	Successfully Tx packets	Practical Throughput
1000	1	0.01
1500	2	0.02
2000	52	0.52
3000	52	0.52
4000	52	0.52

Model Graph:



Experiment 7 – STOP & WAIT WITH BER

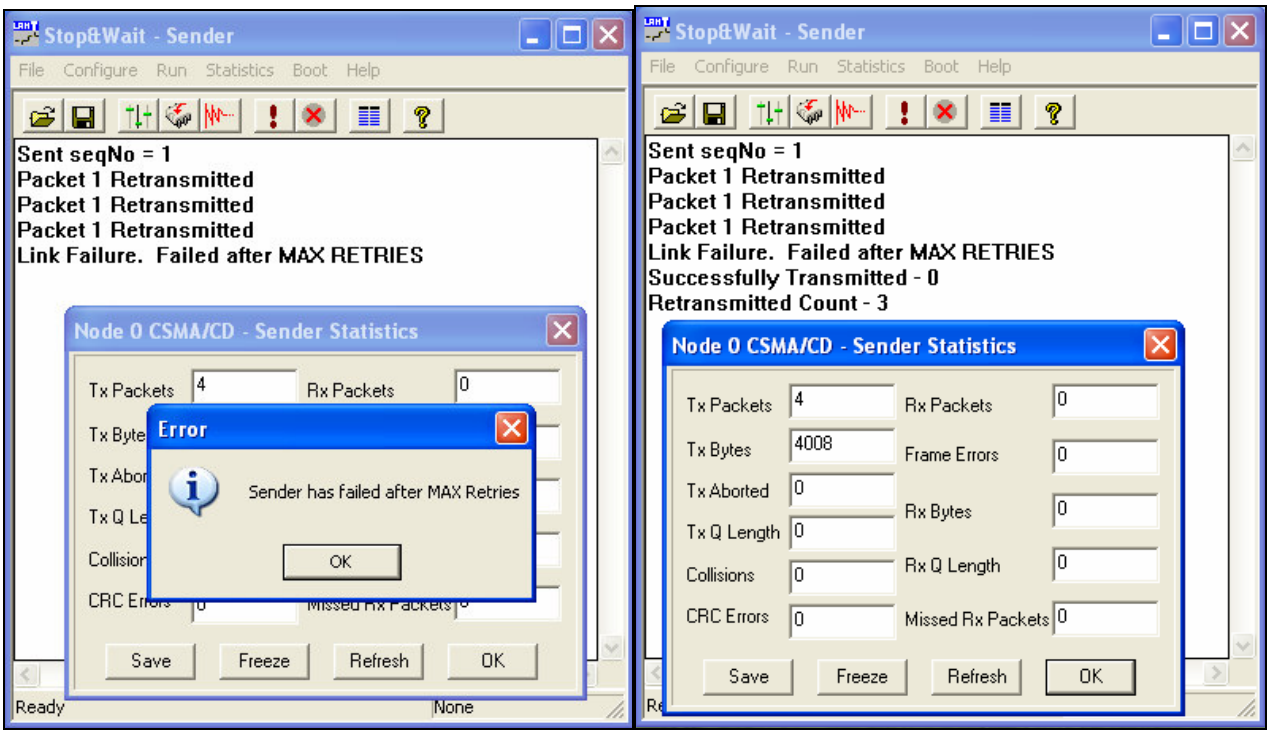
Aim:

Provide reliable data transfer between two nodes over an error network using the stop-and-wait protocol

Procedure:

1. Set the error rate to 10^{-2} in the **NEU**.
2. Follow the **Stop and Wait** Experiment procedure for running the experiment.
3. Set the **time out** value as 3000 ms in the sender window.

PC 1 SENDER **PC 2 RECEIVER**



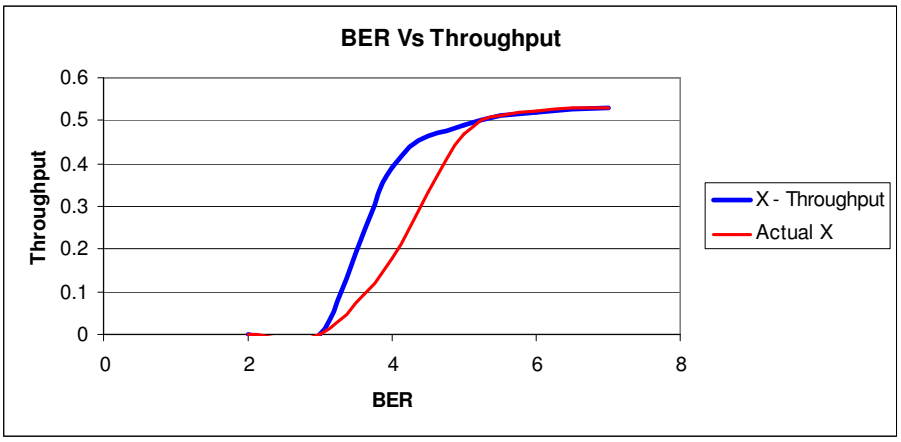
4. From the Statistics window note down the number of successfully transmitted packets and calculate the throughput. (Calculation of throughput is same as explained in the previous expt.)
5. Repeat the Experiment by setting different BER in the NEU.
6. Use the values to plot the graph between BER Vs Throughput

Note: You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template

Model Tabulation:

BER	Successfully Tx packets	Theoretical Throughput	Practical Throughput
10^{-6}	53	0.53	0.53
10^{-5}	52	0.52	0.5232
10^{-4}	9	0.49	0.4675
10^{-3}	40	0.39	0.1787
10^{-2}	0	0	0

Model Graph:



Experiment 8 – SLIDING WINDOW GO BACK N

Aim:

Provide reliable data transfer between two nodes over an unreliable network using the sliding window go back N protocol

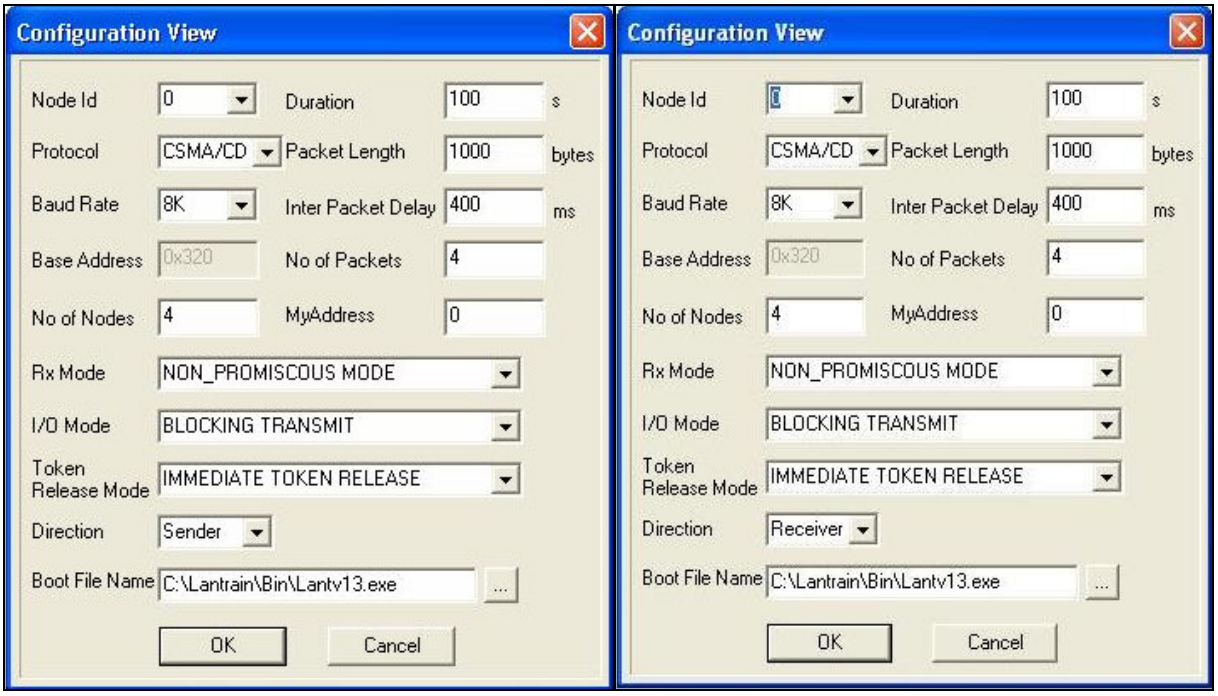
Procedure:

1. Click on the **Sliding Window GBN** icon

from the desktop on both PCs.

2. Click the Configuration button in the window in both the Pc's.

PC 1 SENDER **PC 2 RECEIVER**



Setting the configuration menu:

PC 1		PC 2	
Node id	**0**	**Node id**	**0**
Protocol	CSMA/CD	**Protocol**	CSMA/CD
Baud Rate	8Kbps (At both the config menu and NEU)	**Baud Rate**	8Kbps (At both the config menu and NEU)
Duration	100s	**Duration**	100s
Packet Length	1000 bytes	**Packet Length**	1000 bytes
Bit Delay	0(at NEU)	**Bit Delay**	0(at NEU)
Direction	Sender	**Direction**	Receiver
No of packets	4	**No of packets**	4

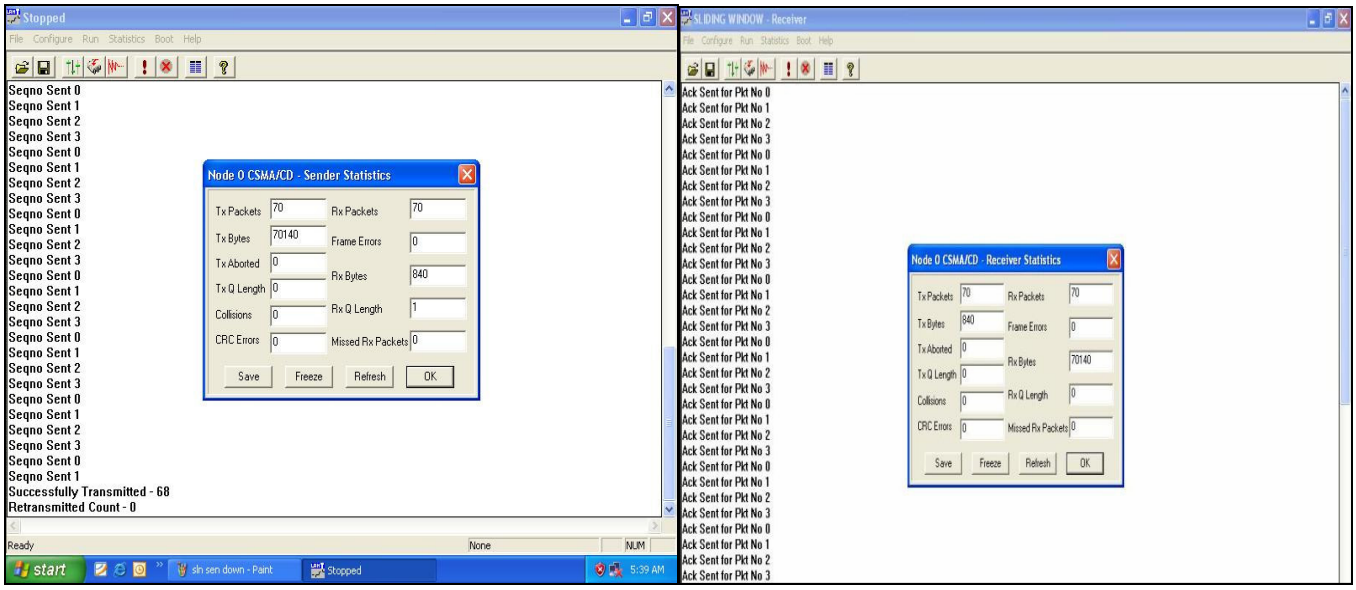
Note: The No of Packets parameter defines the window size.

3. Set the **Inter Packet Delay** to 400msecs

4. Click **OK** button and Download the driver to the NIU using the BOOT button command. Booting from any one of the applications is enough.

5. Run the experiment by clicking button or by choosing RUN → Start from each application.
6. Set the Timeout Value to **1500** ms

PC 1 SENDER **PC 2 RECEIVER**



7. Note down the no of successfully Transmitted Packets.
8. Repeat the above steps for various time out values and plot the graph between timeout Value & Throughput. Find the optimum timeout value from the plot.

Note: You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template

9. Explain why the throughput is less compared to CSMACD protocol.

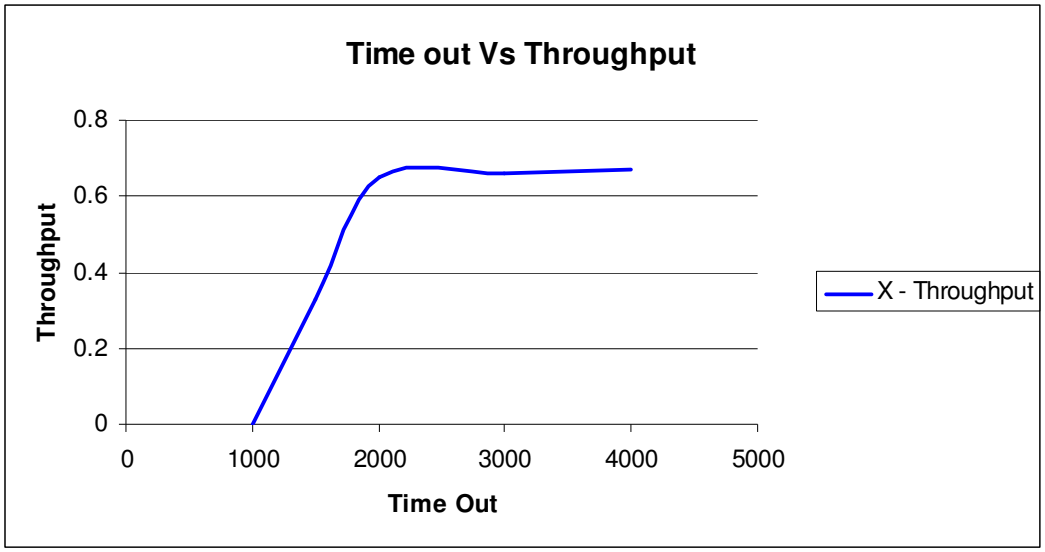
Calculation of Practical Throughput:

$$X = \frac{\text{(Sum of Successfully Tx packets * Packet Length * 8)}}{\text{(Duration of Experiment * Data rate)}}$$

Model Tabulation:

Time out value in ms	Successfully Tx packets	Practical Throughput
1000	0	0
1500	33	0.33
2000	65	0.65
3000	66	0.66
4000	67	0.67

Model Graph:



Experiment 9 – SLIDING WINDOW GO BACK N WITH BER

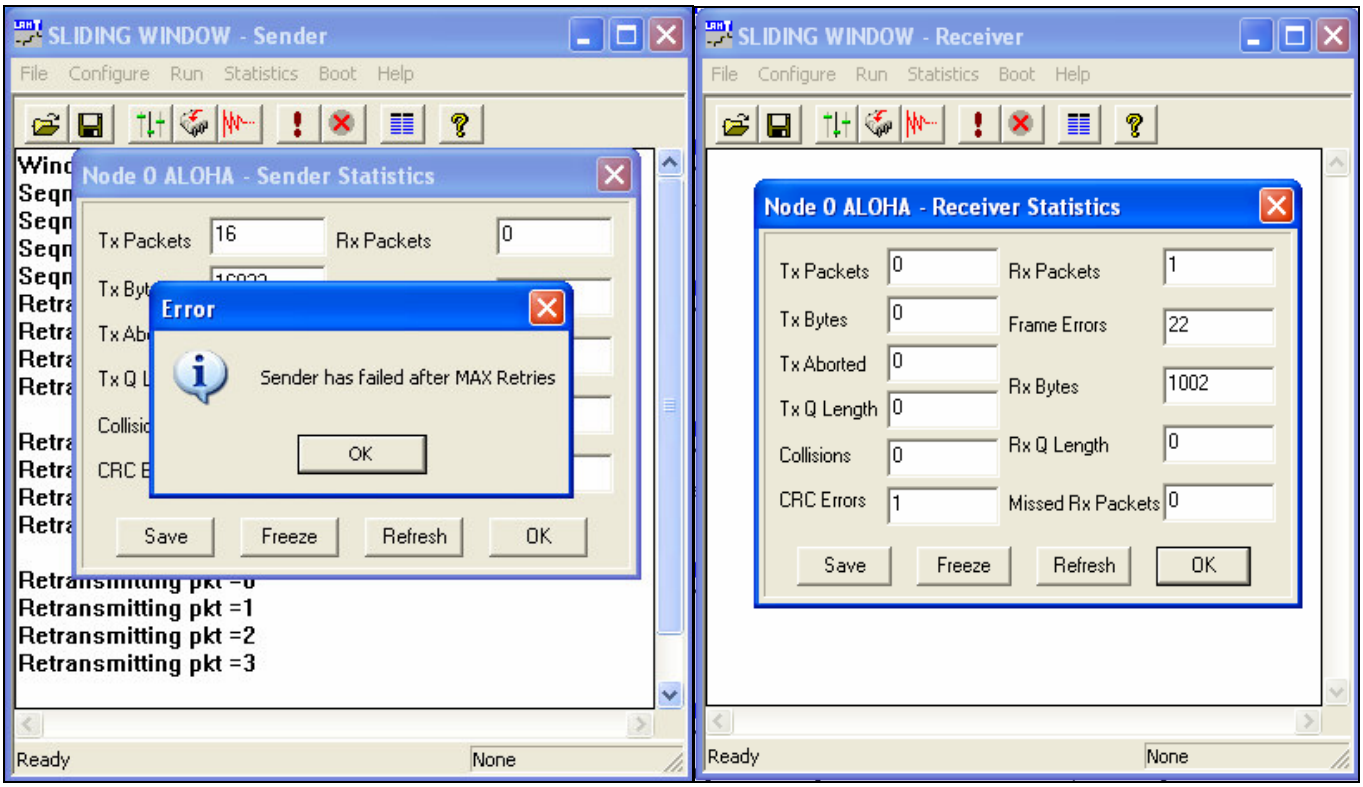
Aim:

Provide reliable data transfer between two nodes over an error network using the sliding window GBN protocol

Procedure:

1. Set the error rate to **10^{-2}** in the **NEU**.
2. Follow the **Sliding Window GBN** experiment procedure for running the experiment.
3. Set the **time out** value as **3000** ms in the sender window.

PC 1 SENDER **PC 2 RECEIVER**



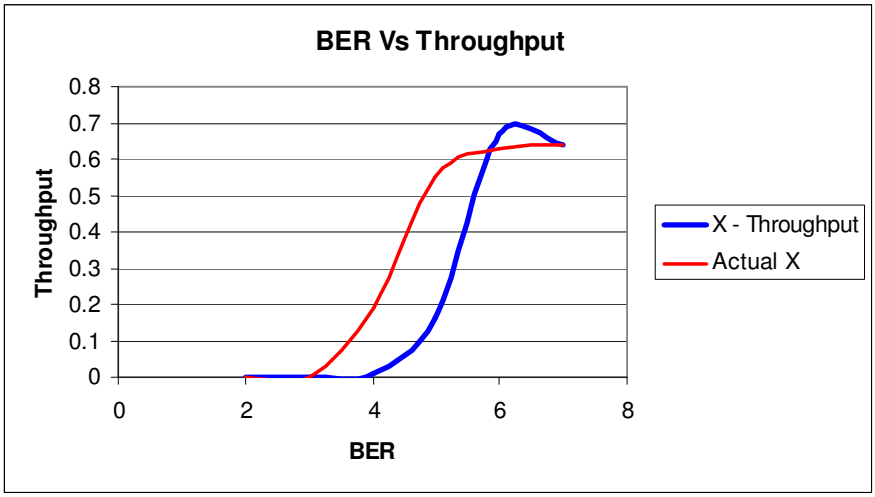
4. From the Statistics window note down the number of successfully transmitted packets and calculate the throughput. (Calculation of throughput is same as explained in the previous expt.)
5. Repeat the Experiment by setting different BER in the NEU.
6. Use the values to plot the graph between BER Vs Throughput

Note: You can also use the template for plotting the graph. Please refer *Appendix-1* to plot the graph using the template

Model Tabulation:

BER	Successfully Tx packets	Theoretical Throughput	Practical Throughput
10^{-6}	53	0.63	0.67
10^{-5}	52	0.55	0.16
10^{-4}	9	0.18	0.01
10^{-3}	40	0	0
10^{-2}	0	0	0

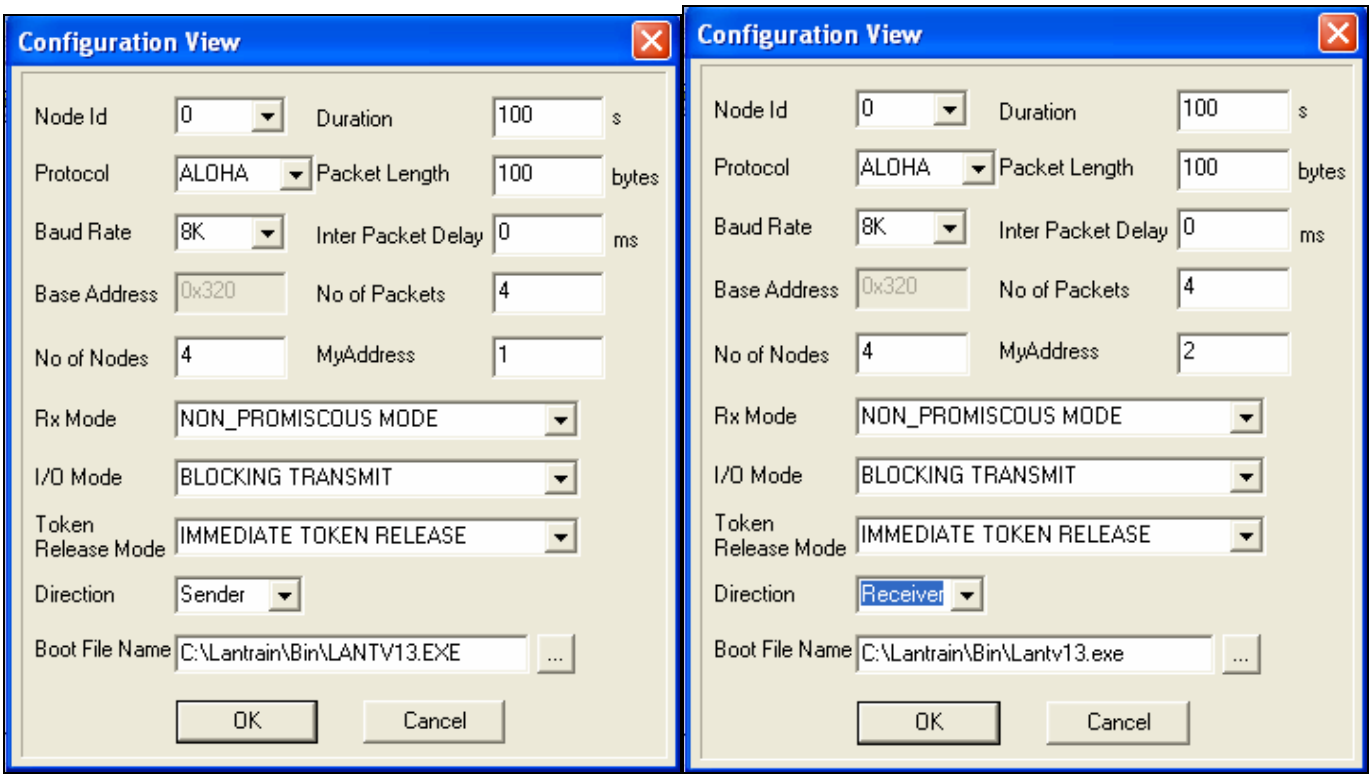
Model Graph:



Experiment 10 – PACKET TRANSMISSION

1. Click on the **Pkt** icon from the desktop of both the PC's.
2. Click the Configuration button in the window in both the PC's.

PC 1 SENDER **PC 2 RECEIVER**

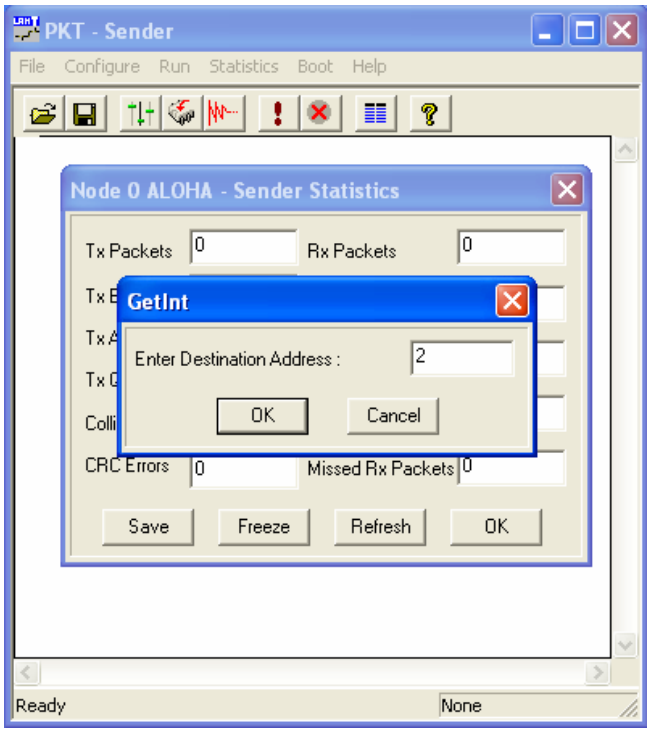


Setting the configuration menu:

PC 1		PC 2	
Node id	**0**	**Node id**	**0**
Baud Rate	8Kbps (At both the Config menu and NEU)	**Baud Rate**	8Kbps (At both the config menu and NEU)
Duration	100s	**Duration**	100s
Packet Length	100 bytes	**Packet Length**	100 bytes
Bit Delay	0(at NEU)	**Bit Delay**	0(at NEU)
Direction	Sender	**Direction**	Receiver

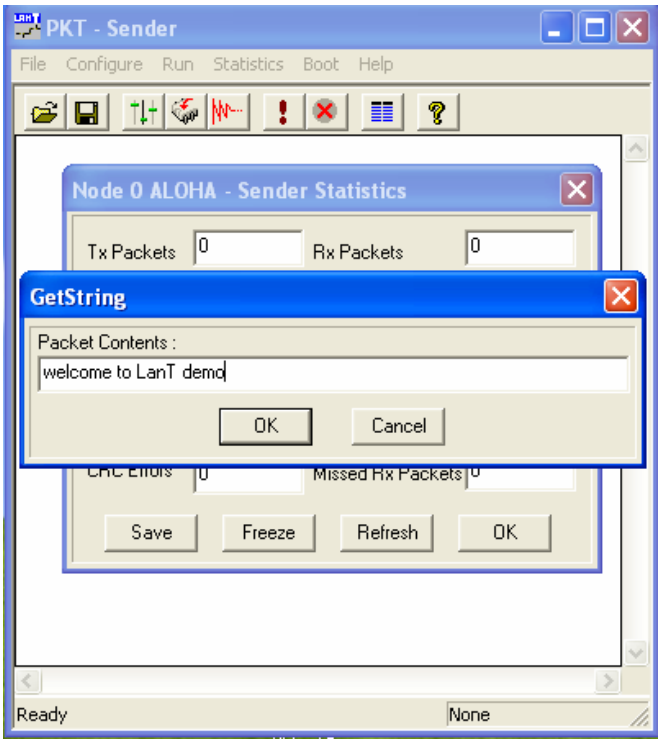
3. Download the driver to the NIU using the BOOT button command for both PCs.
4. Run the Receiver node first & then the transmitter window. Enter the receiver's node ID in the sender's dialogue box, once it is prompted.

PC 1 SENDER



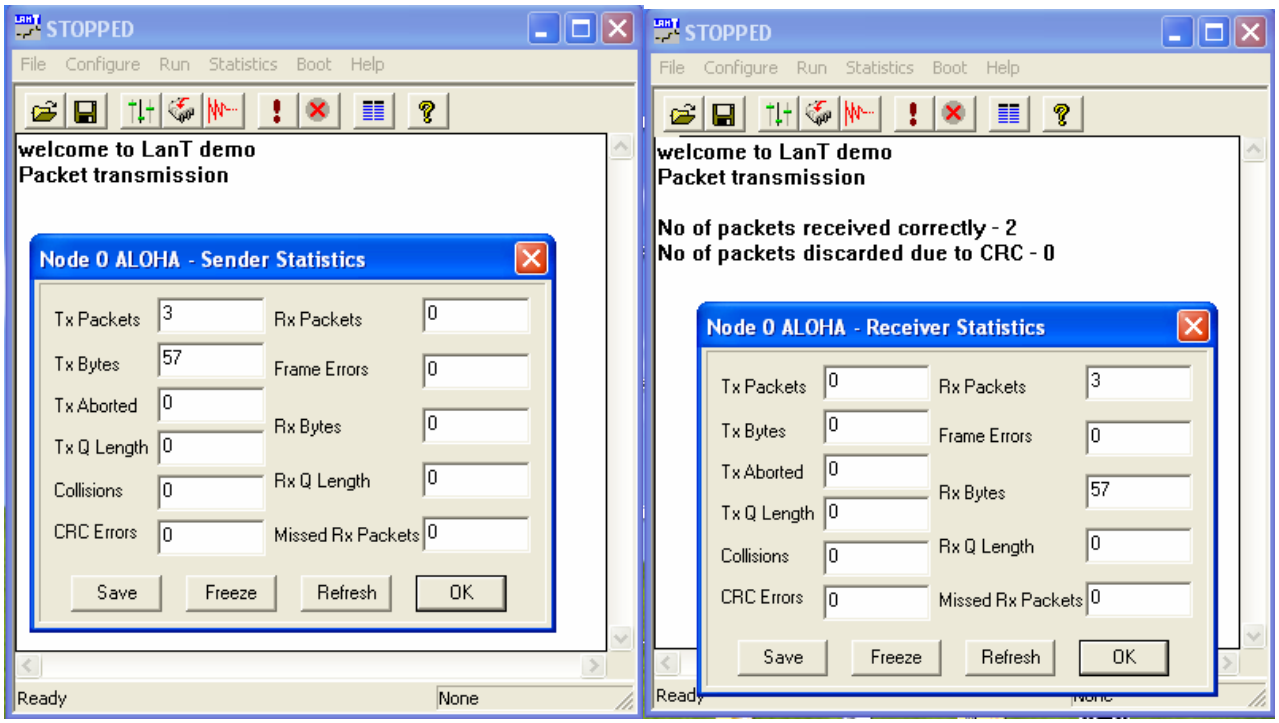
5. After this the sender will display another **Get String** dialogue box, here you can key in your string using the keyboard & press the ok button.

PC 1 SENDER



6. Observe the text in the receiver node. If an empty message is sent, both the sender and receiver should terminate.

PC 1 SENDER **PC 2 RECEIVER**

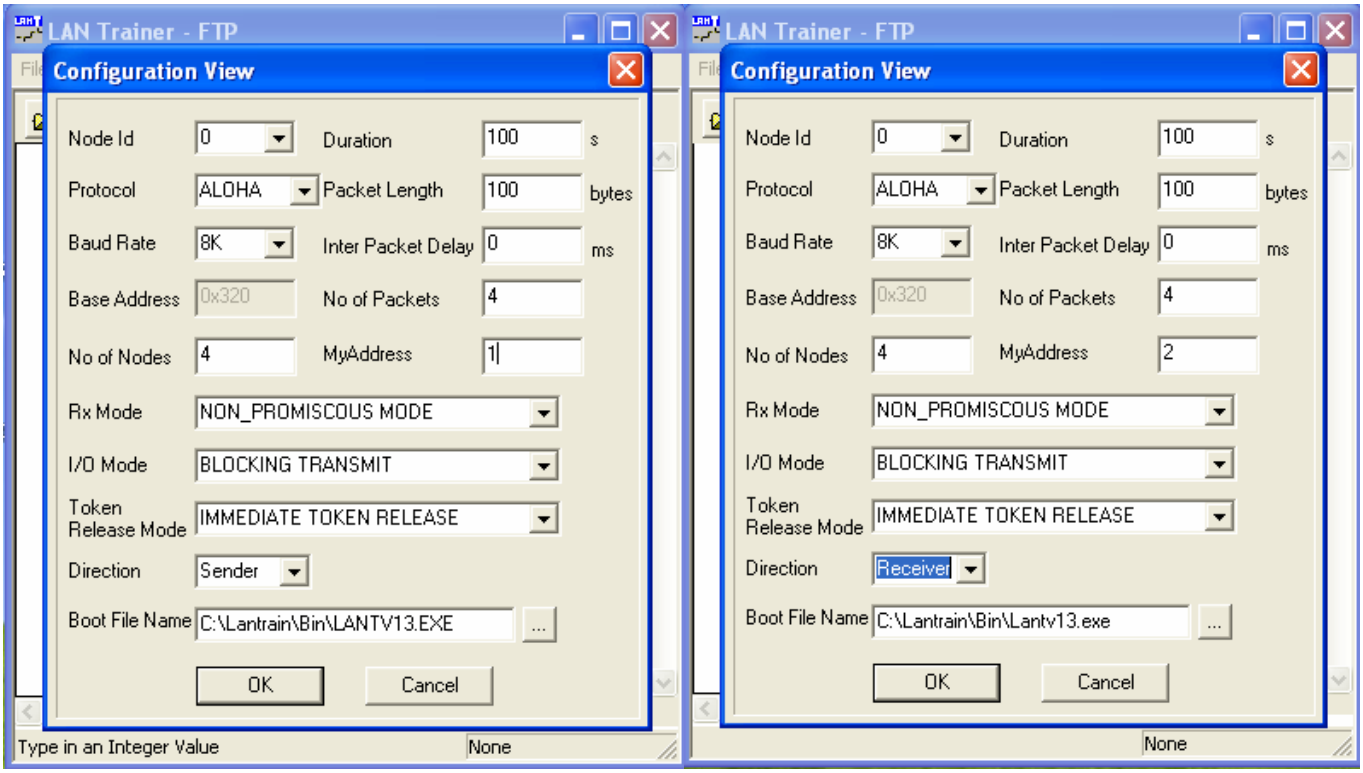


Set the BER to **10^{-4}** and repeat the above-mentioned steps and observe the results.

Experiment 11 – FTP

1. Click on the **FTP** icon from the desktop on both PCs.
2. Click the Configuration button in the window in both the PC's.

PC 1 SENDER **PC 2 RECEIVER**


LAN Trainer - FTP
Configuration View
Node Id 0 Duration 100 s
Protocol ALOHA Packet Length 100 bytes
Baud Rate 8K Inter Packet Delay 0 ms
Base Address 0x320 No of Packets 4
No of Nodes 4 MyAddress 1
Rx Mode NON_PROMISCOUS MODE
I/O Mode BLOCKING TRANSMIT
Token Release Mode IMMEDIATE TOKEN RELEASE
Direction Sender
Boot File Name C:\Lantrain\Bin\LANTV13.EXE
OK Cancel
Type in an Integer Value
None
LAN Trainer - FTP
Configuration View
Node Id 0 Duration 100 s
Protocol ALOHA Packet Length 100 bytes
Baud Rate 8K Inter Packet Delay 0 ms
Base Address 0x320 No of Packets 4
No of Nodes 4 MyAddress 2
Rx Mode NON_PROMISCOUS MODE
I/O Mode BLOCKING TRANSMIT
Token Release Mode IMMEDIATE TOKEN RELEASE
Direction Receiver
Boot File Name C:\Lantrain\Bin\Lantv13.exe
OK Cancel
None

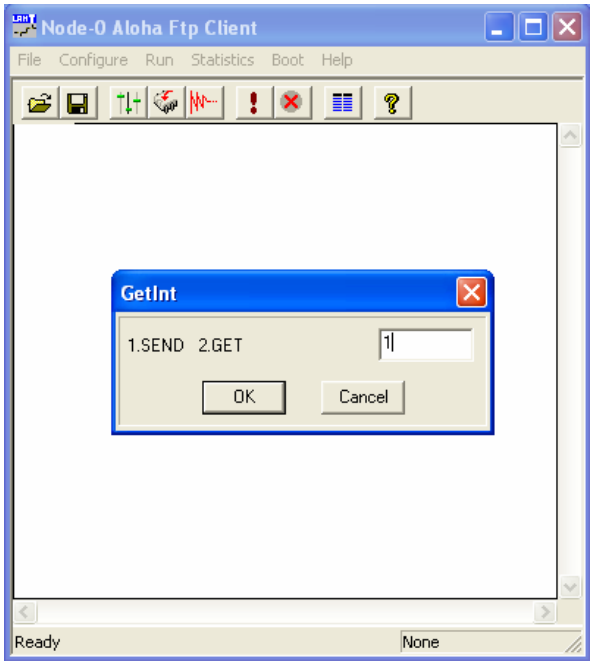
Setting the configuration menu:

PC 1		PC 2	
Node id	**0**	**Node id**	**0**
Baud Rate	8Kbps (At both the config menu and NEU)	**Baud Rate**	8Kbps (At both the config menu and NEU)
Duration	100s	**Duration**	100s
Packet	100 bytes	**Packet**	100 bytes
Bit Delay	0(at NEU)	**Bit Delay**	0(at NEU)
Direction	Sender	**Direction**	Receiver

3. Set the IPD as 40 ms.
4. Download the driver to the NIU using the BOOT button command for both PCs.

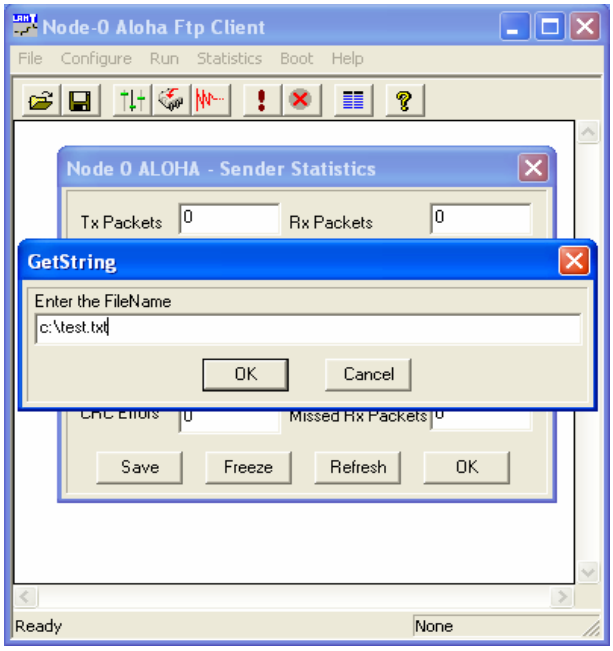
5. Run the Receiver node first & then the transmitter window. While you do this, the sender node will display a **GetInt** message box, if you want to transmit a file from Client to server, Enter 1 or else if you want to download a file from Server to client, press 2.

PC 1 SENDER



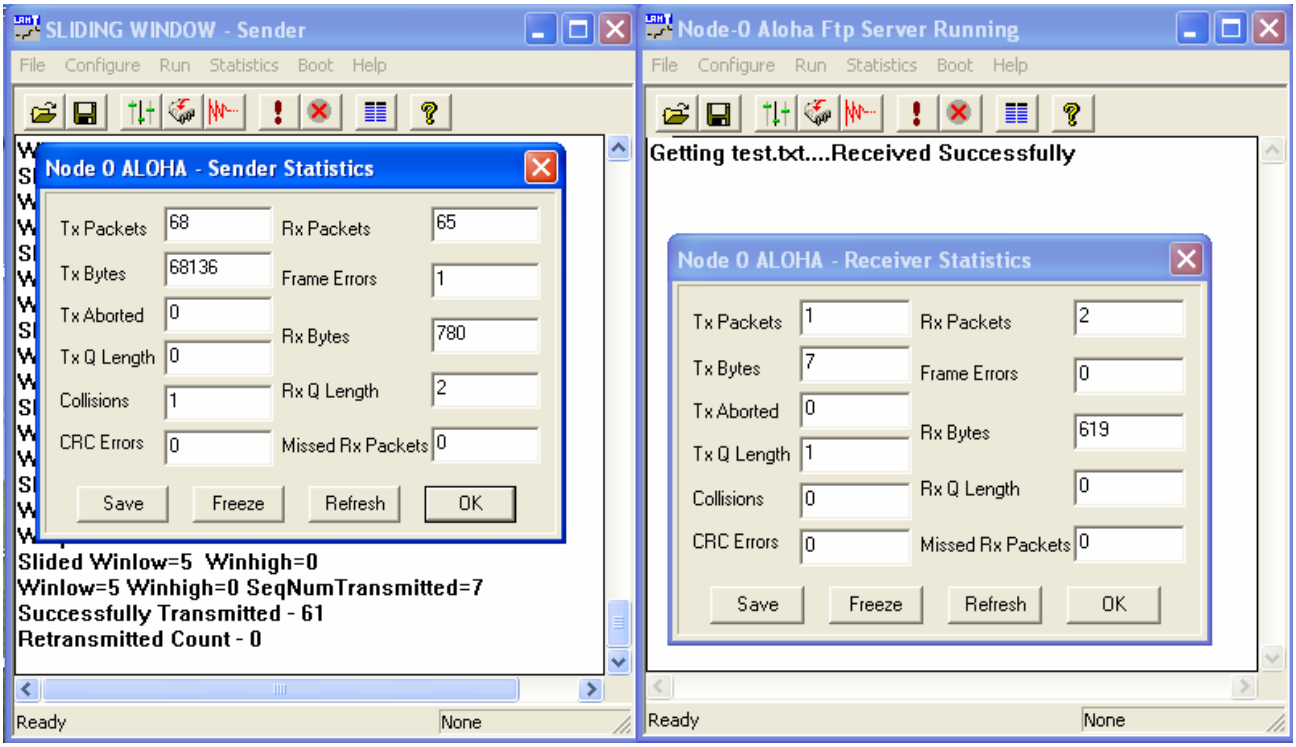
6. Then Specify the Path of the file either to transmit or to download (i.e. c:\test.txt) & specify the timeout value.

PC 1 SENDER



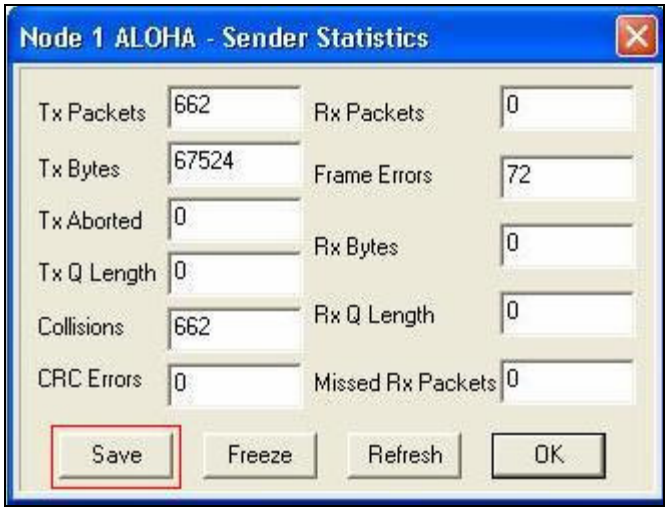
7. At the end of transmission, the file will be stored in the server node's desktop.

PC 1 SENDER **PC 2 RECEIVER**



Procedure to plot graph for MAC Layer experiments using MS EXCEL template

Save the readings of the **MAC** layer experiments by clicking **Save** button from the **Statistics** window in all the nodes.



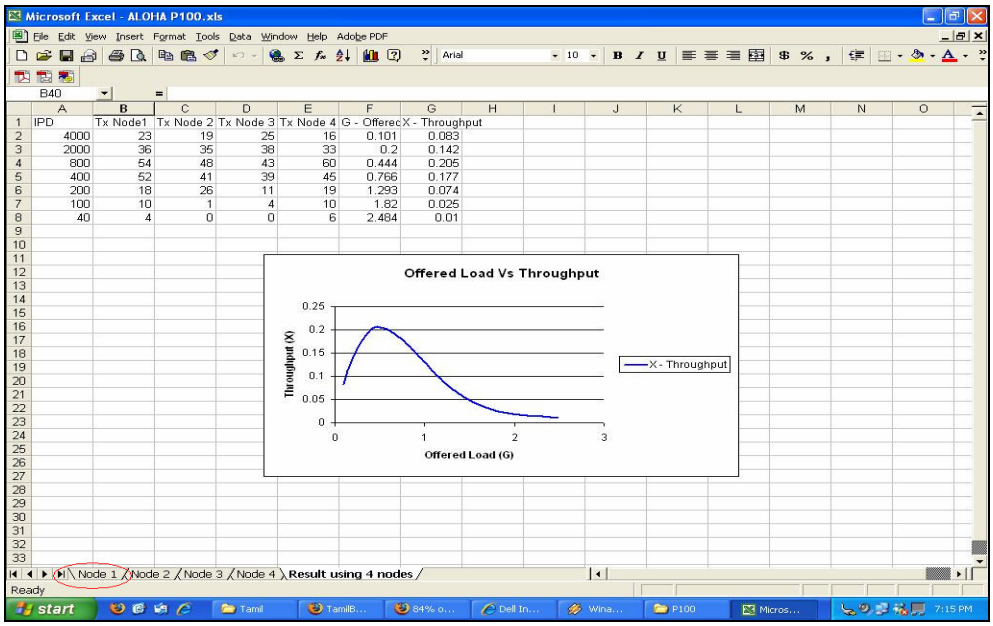
By default the file get saved in the Desktop. (You can also save it in other location).

Organise all the files in a common location.

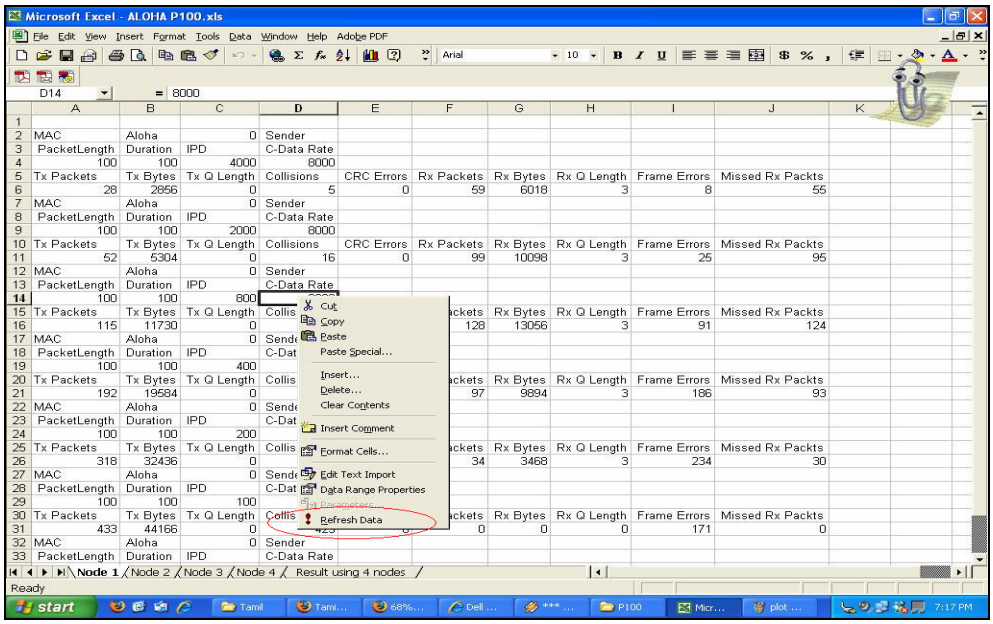
Open **MS EXCEL**.

Browse open **.xls** file in **c:\Lantrain\Stat....** (Say:\Lantrain\Stat\Aloha\P100\ALOHAP100.xls).

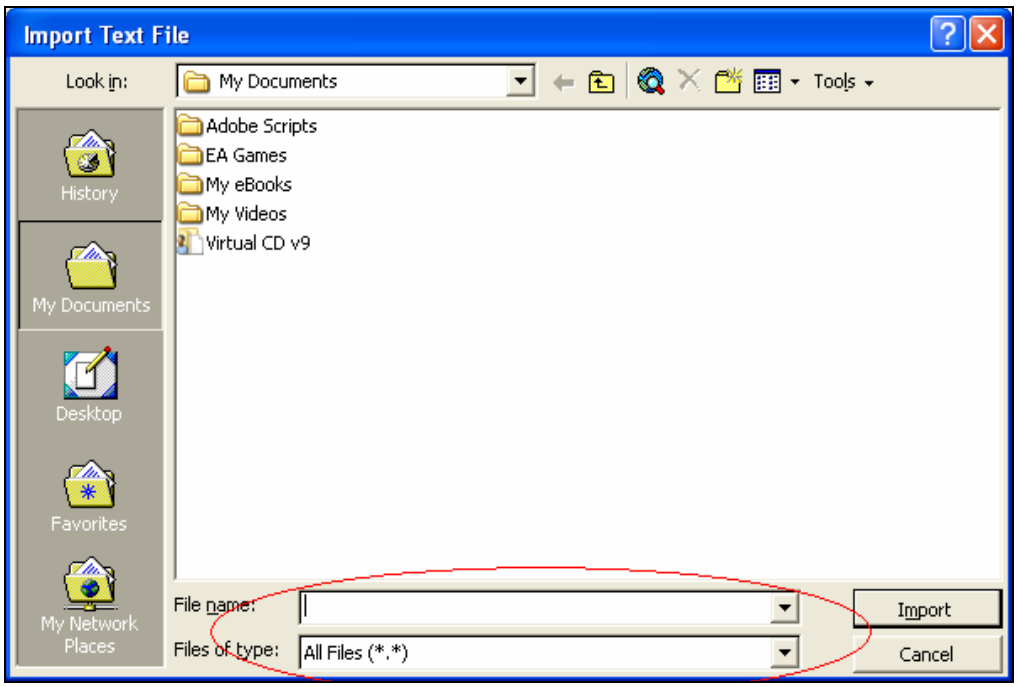
Click **Node1** sheet.



Go to **Data** → **Refresh data**.

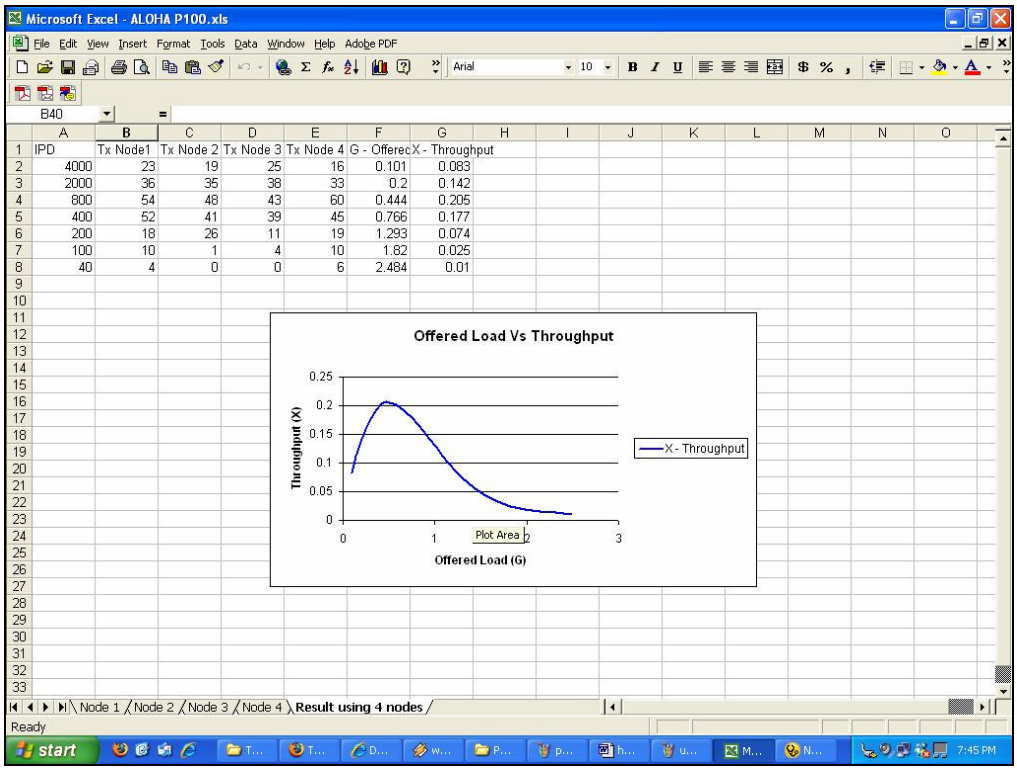


Browse the respective file to be replaced with existing data for **Node1**.



Repeat the above step for all the four nodes.

Now click the **RESULT USING 4 nodes** sheet to view the graph.

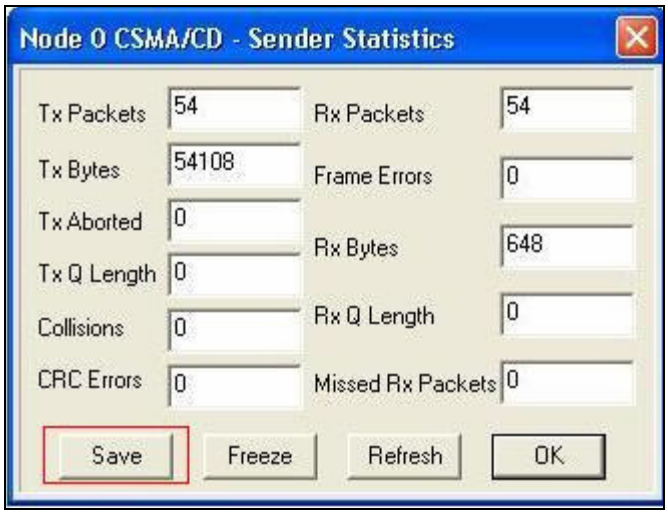


	A	B	C	D	E	F	G
1	IPD	Tx Node1	Tx Node 2	Tx Node 3	Tx Node 4	G - Offerec	X - Throughput
2	4000	23	19	25	16	0.101	0.083
3	2000	36	35	38	33	0.2	0.142
4	800	54	48	43	60	0.444	0.205
5	400	52	41	39	45	0.766	0.177
6	200	18	26	11	19	1.293	0.074
7	100	10	1	4	10	1.82	0.025
8	40	4	0	0	6	2.484	0.01

Note: Save exact seven readings for **MAC** experiments while using this template. Shortage or excess number of readings may produce errors.

Procedure to plot graph for MAC Layer experiments using MS EXCEL template

Save the readings of the **LLC** layer experiments by clicking **Save** button from the **Statistics** window from the sender and receiver nodes.



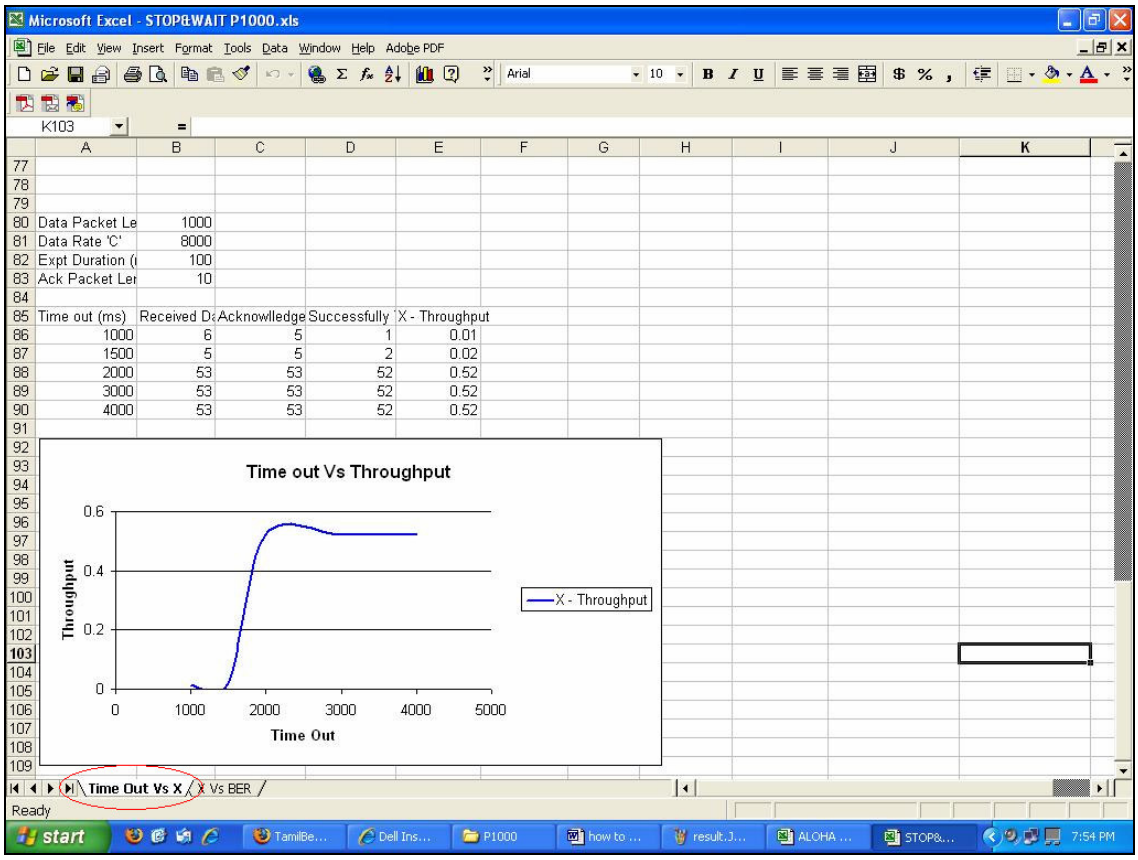
By default the file get saved in the Desktop. (You can also save it in other location).

Organise all the files in a common location.

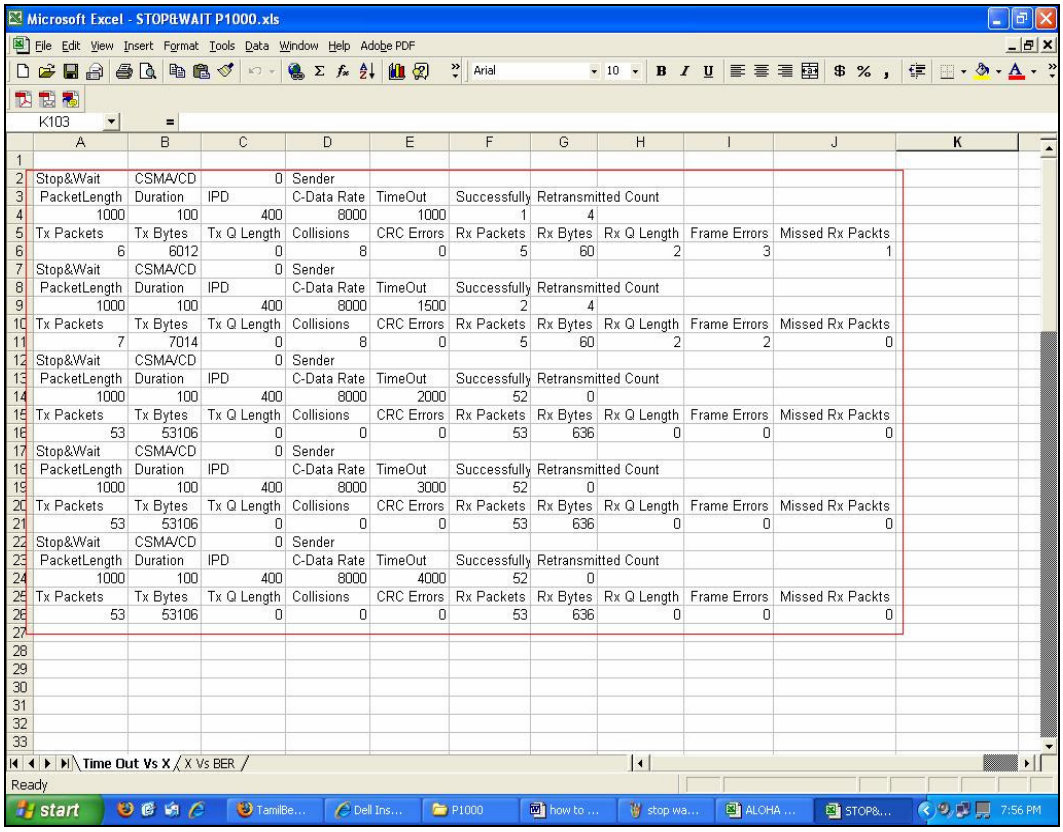
Open **MS EXCEL**.

Browse open **.xls** file in **c:\Lantrain\Stat....** (Say:\Lantrain\Stat\Stop and Wait\P1000\Stop and wait P1000.xls).

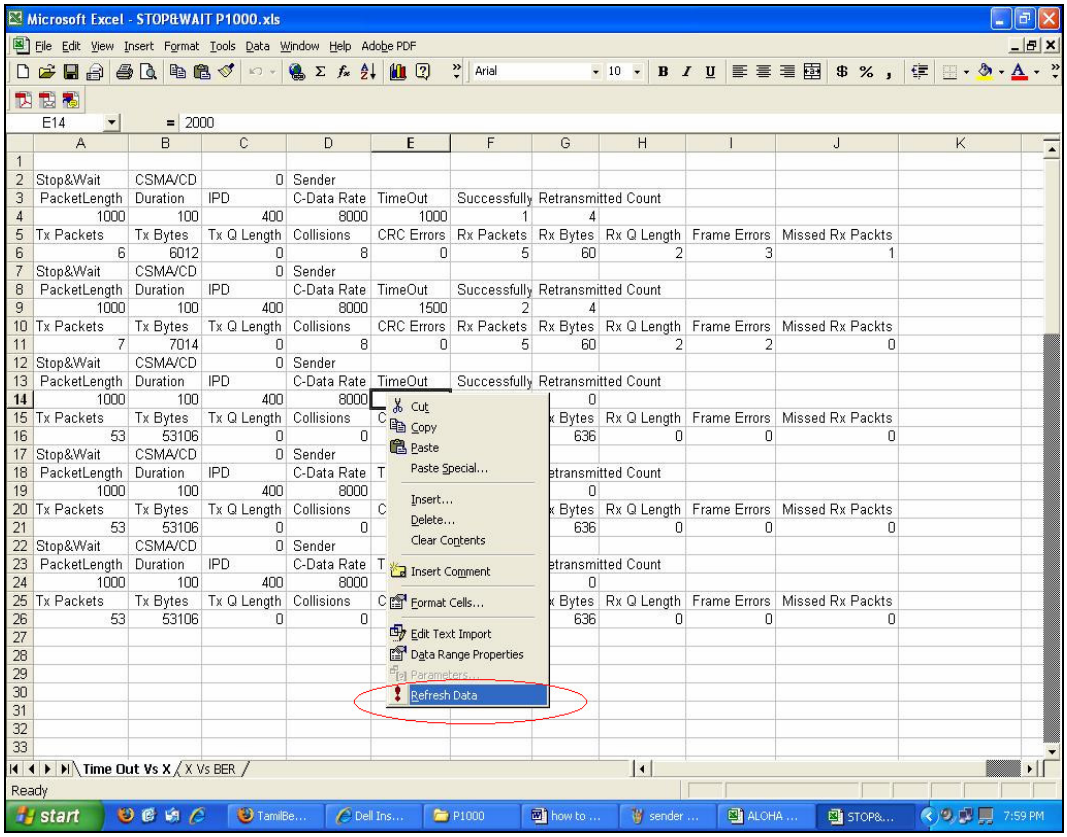
Click **Timeout Vs X** sheet.



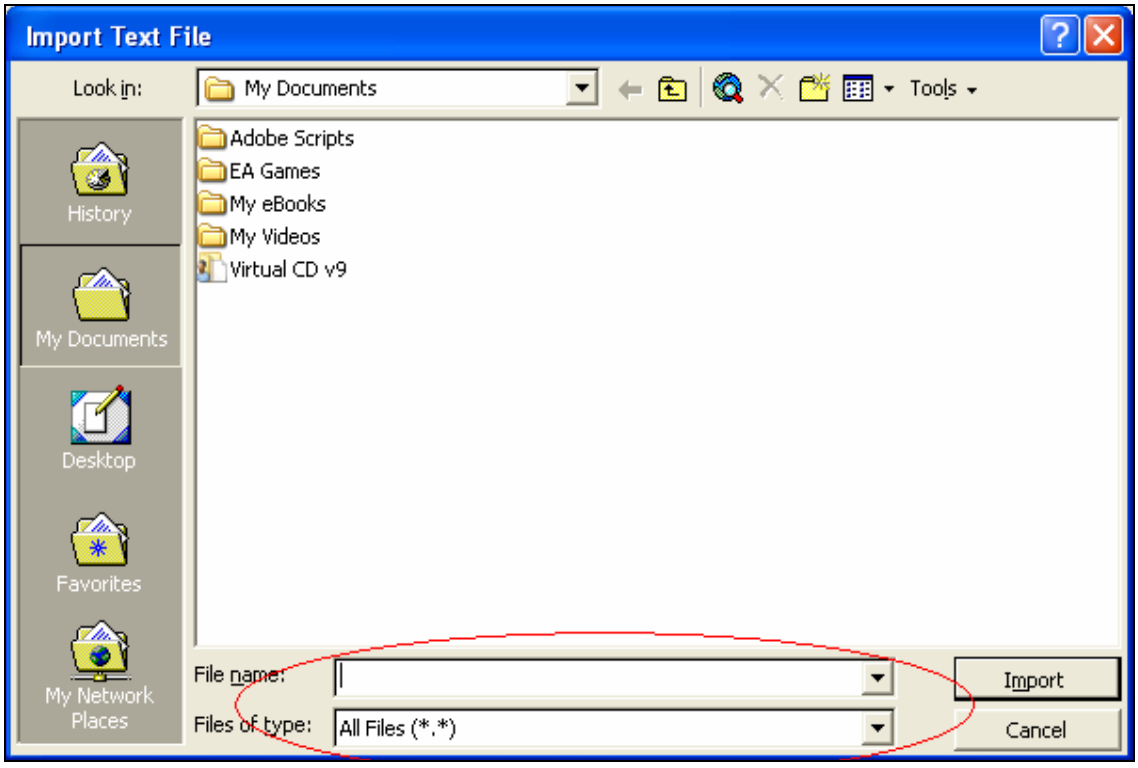
Scroll up to the sender readings.



Go to **Data → Refresh data** option.

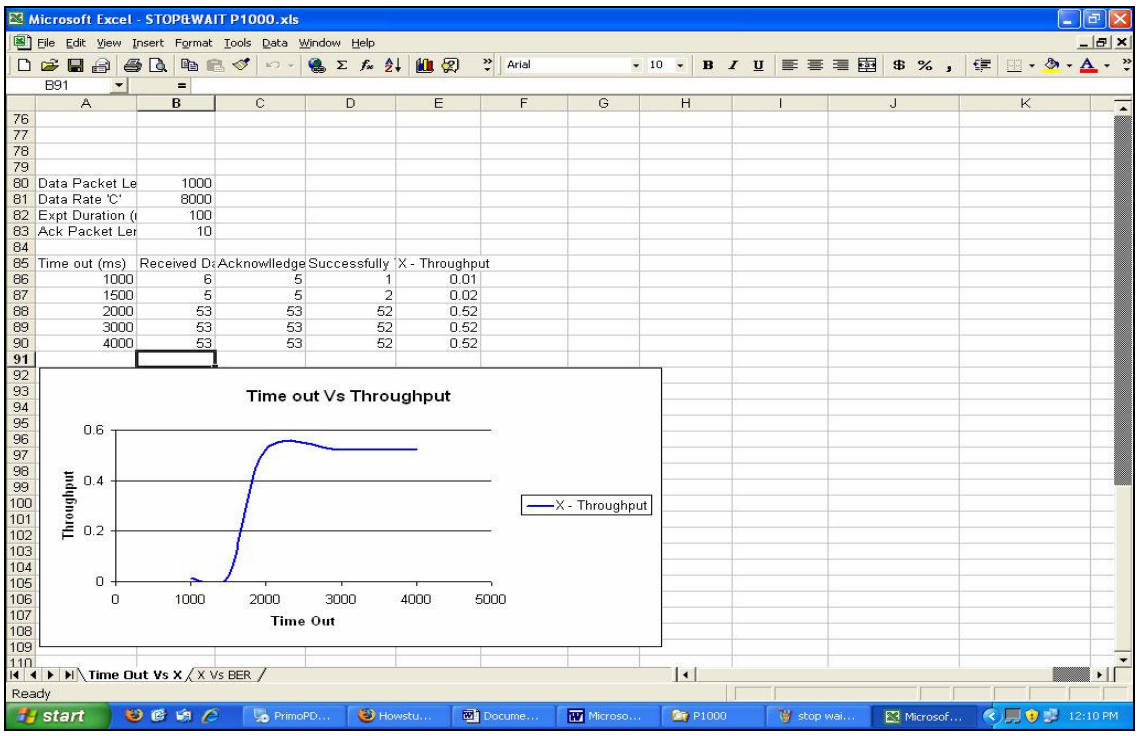


Browse the respective file to be refreshed for the existing data.

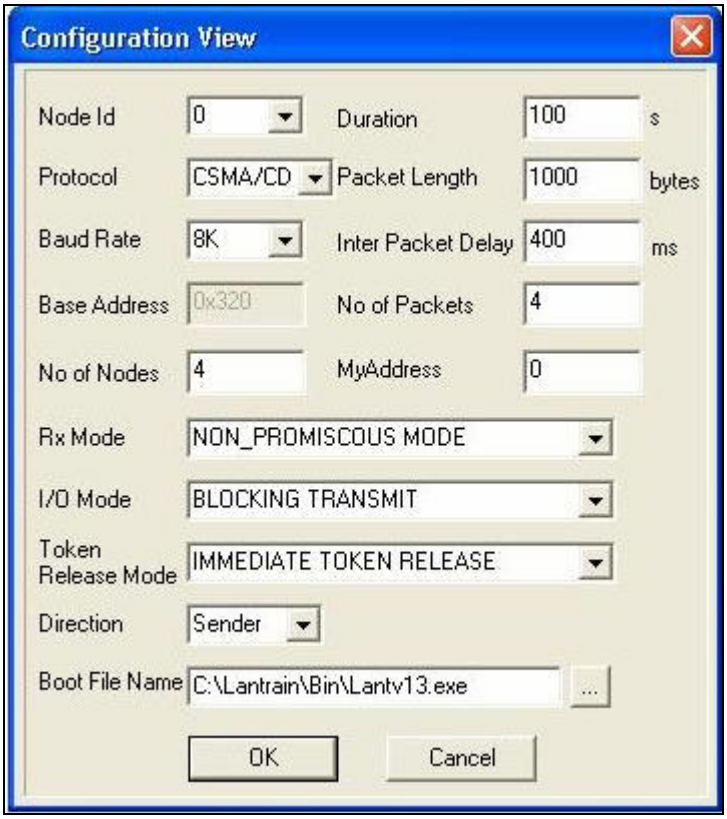


Repeat the same to refresh the existing Receiver data with your data.

Scroll down to see the plot



9. Follow the same procedure to **Plot BER vs X.**



Config View displays the configuration dialog box where all the parameter values are displayed.

Node Id - Specify that that particular application/experiment window should be as node 0 or 1. This is the one that differentiates the two applications in the same PC.

Protocol - Specify the MAC Protocol that NIU has to emulate. By default, it is set to Aloha

Baud Rate - Specify the data rate between 8Kbps and 1Mbps. (Only for CSMA/CD and Ring mode, this needs to be set both in the Application window and the NEU. For Aloha and CSMA, settings in NEU are sufficient.)

No of Nodes - Specify the number of nodes in the network that are used to do Token Ring experiment. This field may be used for other purposes while writing codes for suggested experiments in exercise also.

Duration - Specify the duration of the experiment in seconds.

Packet Length - Specify the length of packets that are used in the experiment. Can be set to a maximum of 1000 bytes.

Inter Packet Delay - Specify the inter packet delay (that is equivalent to the inter packet arrival time) based on the calculation suggested in the experiments. This has to be in milliseconds.

Number of Packets - Specify number of packets in a window for Sliding Window experiment. This field may be used for other purposes while writing codes for suggested experiments in exercise, say for example, to specify number of packets to be transmitted or received in a MAC experiment instead of following time.

My Address - Specify the address of each node (between 1 and 6) in top layers. For example, the server uses this field in FTP to identify the client for sending and receiving files.

Rx Mode - Specify Promiscous or Non-Promiscous mode of packet reception. By default all the stand-alone experiments use promiscuous mode of reception (receives all the packet). Altering this field will not have effect in the stand-alone experiments. Refer programmer's manual for using various types while writing codes for exercises.

I/O Mode - Fixes the types of transmit and receive modes. In majority of the stand-alone experiments, blocking transmit and receive is used and the other types are left for the user to implement wherever necessary as suggested in exercises. Altering this field will not have effect in the stand-alone experiments. Refer programmer's manual for using various types while writing codes for exercises.

Token Release Mode - Specify the type of token release mode to be followed in token ring. Manual token release is used in token ring experiment. Altering this field will not have effect in this stand-alone experiment. Refer programmer's manual for using various types while writing codes for exercises.

Direction - Specify which direction to be followed by each node. In MAC experiments, Sender will only transmit data packets and Receiver will only receive. In DLL experiments, Sender will transmit data packets and receive acknowledgement packets while Receiver will receive them and transmit acknowledgement packets. In FTP Sender is the client and Receiver is the server.

Boot File Name - Name and path of the NIU driver file. This is for future use. The default file and location is **C:\Lantrain\bin\LantV13.exe**. It may be left to this default name.

Developing LAN Trainer Applications

The LAN Trainer API is a Windows DLL (Dynamic Link Library) written in C. Microsoft Visual C++ compiler that comes under Microsoft Visual Studio is used for the Shell and the experiments. Microsoft Visual Studio provides a development environment for Visual C++. This integrated set of tools runs under Windows 95, 98, Windows NT or Windows 2000 or other operating systems as recommended by Microsoft. The Developer Studio user-interface consists of an integrated set of windows, tools, menus, toolbars, directories, and other elements that allow you to create, test, and refine your application in one place.

This Appendix describes the organization of the source code into workspaces and projects and how you can use the LAN Trainer to write network software. You can modify the source code of the experiments supplied with the LAN Trainer or implement different protocols and projects by creating them as new experiments. Finally, if you wish, you can customize the LAN Trainer shell.

Organisation of LAN Trainer Source Code

LAN-T software uses a single workspace for all the experiments and therefore all experiment projects are created under the workspace called *Expts.dsw* that will be in *C:\Lantrain\Src* directory.

A workspace is the one that contains one or more projects. A project is the one that generates the executable file (ex: MAC.exe). In LAN-T software, *Expts.dsw* is the workspace and the experiments are the projects namely, *MAC* (for ALOHA, CSMA, CSMA/CD), *TknBus, TknRing, Pkt, SlWinGBN, StopWait, FTP*.

Modifying an Existing Experiment

MAC experiment is taken here as an example to list the steps involved in modifying an existing experiment. Each experiment is a project under the workspace discussed above.

- Open experiment workspace from the following path *C:\Lantain\Src\Expts.dsw* in VC++
- Make the project that you want to modify, say MAC, active using the option *Set Active Project* from the *Project* menu.
- Open *MAC.cpp* from the *Source* folder and do the necessary coding.
- Save workspace using *Save Workspace* menu option under *File* menu and *Build* the project (*Build* is the process of compilation and linking that creates the *.exe file that is ready to run) using the key F7. Once the *Build* result is successful run the experiment by pressing the keys Ctrl+F5. Please refer Microsoft VC++ User Manual for more options in compiling and linking a project file.

Creating a New Experiment

- Open experiment workspace from the following path *C:\Lantrain\Src\Expts.dsw* in VC++
- Create a new project by clicking New in the File menu.
- Choose a project name (say *Test*) and choose *MFC Appwizard (EXE)* as project type. Check the option *Add to current workspace* and finally click the OK button.
- Select *Single document* and click Finish button. Now the wizard will create a Project Workspace (*Test*) with three directories namely *Source Files* (containing *.cpp files), *Header Files* (containing *.h files) and *Resource Files* (containing resource files) under it. This project will now be the active project as far as VC++ workspace is concerned.
- Delete all the files from this project workspace i.e. from *Source Files*, *Header Files* and *Resource Files* directory. [To delete a file from workspace click that file and press Del key.]
- Open Windows Explorer and delete all files from the working directory (*C:\Lantrain\Src\Test*) including the *Res* directory except the **.dsp* file – in our example it is *Test.dsp* file.
- Copy the following files from *C:\Lantrain\Src\ShellVxy* to the current project directory *C:\Lantrain\Src\Test*

- *Res* directory.
- *stdAfx.cpp* and *stdAfx.h*
- *resource.h* and *resource.hm*
- *lanT.rc* and *lanT.aps*
- A sample *mac.cpp* file
- Rename this *mac.cpp* file as *Test.cpp*
- Note: The directory and filename Shellxy represents the LAN-T Shell with Version specified by x and y. For example SHellV13 denotes LAN-T Shell Ver1.3

- Above said files have to be added in the project workspace under the appropriate folder names. To add, click the folder name (say *Source Files*) and right click the mouse button and select *Add files to Folder*. This will open up the File Open dialog box through which you can add files that are listed below in their appropriate folders.

- Select *Source Files* folder and add the following
    - *lanT.rc*
    - *stdAfx.cpp*
    - *Test.cpp file*
- Select *Header Files* folder and add the following
    - *resource.h*
    - *resource.hm*
    - *stdAfx.h*
- Select *Resource Files* folder and add all the files in the *Res* directory.
- Now save the entire workspace.
    - Under *Projects* menu select *Settings* and do the following:
- Select *General* tab and choose *Use MFC in a static library*.
- Select *Link* tab and link the libraries by writing *Shellvxy.lib Lantrn.lib* in the *Object/Library modules* field.
- Select *C\C++* tab and select the category as *Precompiled headers* and select the *Automatic use of precompiled headers* option.
- Click OK button.
    - Under *Tools* menu select *Options* and select *directories* tab.
- Pull down *Show directories for* and select *include files*. Write *C:\Latrain\Include* as the last entry in the *Directories* field that lists already other required files.
- Pull down *Show directories for* and select *library files*. Write *C:\Lantrain\lib* as the last entry in the *Directories* field that lists already other required files.
- Save the entire workspace and the project is ready for coding and compilation.
- Choose the *Test.cpp* file and do the necessary coding and *Build* it by pressing the key F7 and *run* it by pressing Ctrl+F5.

Customising the *Shell*

The user-interface for all LAN Trainer experiments is provided by a collection of functions referred to as the *Shell*. This method is followed to focus the topic of interest in the algorithm or the method followed in implementing a protocol than on the user-interface part. Even though it is recommended to concentrate on the protocols and its implementation the complete source code for the *Shell* is given in *C:\Lantrain\Src\ShellVxy* directory for the user to customize according to their requirements. The last two alphabets in the filename represent the version number. For example *ShellV13* file is the *Shell* file of version number 1.3. If you wish to customize the *Shell*, follow the steps given below:

- Open experiment workspace from the following path *C:\Lantrain\Src\Expts.dsw* in VC++
- Create a new project by clicking *New* in the *File* menu.
- Choose a project name (say *ShellTest*) and choose *Win32 Static Library* as project type. Check the option *Add to current workspace* and finally click the OK button.
- Click *Finish* button. Now wizard will create a project with two directories namely *Source Files* and *Header Files*. Add a new folder called *Resource Files* by clicking the right button of mouse on the project name (here *ShellTest files*) and selecting *New Folder*. This will open up a dialog box with *Name of the Folder* field and *File extensions* field. In the first field give the name as *Resource Files* and in the second field give **. ** as file extensions.
- Open windows explorer and copy all the files from *C:\Lantrain\Src\ShellVxy* directory to the current project directory *C:\Lantrain\Src\ShellTest*. This current project directory would have been created in the hard disk as soon as the project is added in the workspace as said in the above step.
- Above said files have to be added in the project workspace under the appropriate folder names. To add, click a folder name (say *Source Files*) and right click the mouse button and select *Add files to Folder*. This will open up the File Open dialog box through which you can add files that are listed below in their appropriate folders.

- Select *Source Files* folder and add the following
  - all cpp files *(*.cpp*)
  - *lanT.rc*

- Select Header Files folder and add the following
    - all header files (**.h*)
    - *resource.hm*

- Select *Resource Files* folder and add all the files in the *Res* directory.
    - Under *Projects* menu select *Settings* and do the following:

- Select *General* tab and choose *Use MFC in a static library*.
- Select *C\C++* tab and select the category as *Precompiled headers* and select the *Automatic use of precompiled headers* option.
- Click OK button.
    - Under *Tools* menu select *Options* and select *directories* tab.
- Pull down *Show directories for* and select *include files*. Write *C:\Latrain\Include* as the last entry in the *Directories* field that lists already other required files.
- Pull down *Show directories for* and select *library files*. Write *C:\Lantrain\lib* as the last entry in the *Directories* field that lists already other required files.
- Save the entire workspace and the project is ready for coding and compilation.
    - Shell has all **.cpp* files appropriate for the dialogs used in the user-interface and *mainframe.cpp* for tool bar and menu. You may modify them or add or remove dialogs as you wish and save them.
    - After finishing the coding, *Build* it by pressing the key F7 and it will create a lib file (here *ShellTest.lib*) in the *debug* directory. Copy this lib file from the *debug* directory to *C:\Lantrain\Lib* directory and copy *Ltulib.h* from the shell working directory (here *C:\Lantrain\ShellTest*) to *C:\Lantrain\Include* directory.

Linking the new *Shell* library with experiments

1. 1. Copy the following files from the working directory of the newly created library (here *C:\Lantrain\Src\ShellTest*) to the experiment project directory (say *C:\Lantrain\Src\MAC*) - the one you wish to use with the new user-interface:

   a. a. *Res* directory.

   b. b. *stdAfx.cpp* and *stdAfx.h*

   c. c. *resource.h* and *resource.hm*

   d. d. *lanT.rc* and *lanT.aps*

2) 2) Under *Projects* menu select *Settings* and select *Link* tab and link the library by changing the old library name to new library name alone (say *ShellV13.lib* to *ShellTest.lib*) in the *Object/Library modules* field. Leave the *Lantrn.lib* as unchanged in that field.

3) 3) Select *RebulidAll* from the Project menu to compile the experiment. Now the experiment is ready to run with the new user-interface.